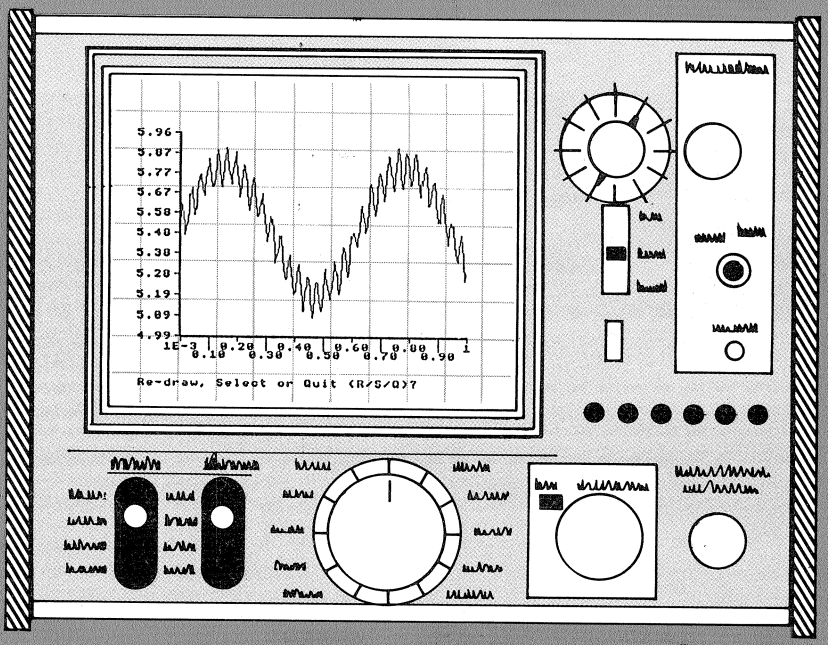
VOLUME 5 Number 5 OCTOBER 1986, PRICE £1.30

BEEBUG

FOR THE BBC MICRO

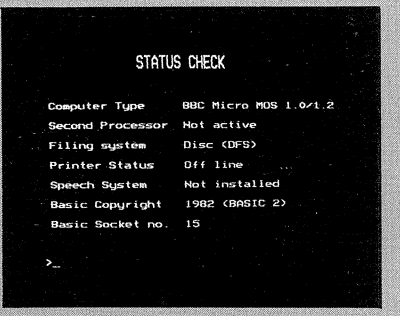


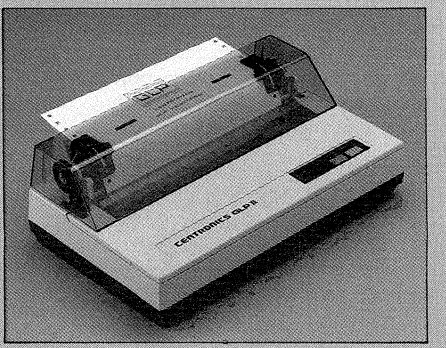
DIGITAL STORAGE OSCILLOSCOPE

Master Compact

Tutankhamen

Which Machine?





Printer Survey

BEEBUG

VOLUME 5 NUMBER 5
OCTOBER 1986

FEATURES

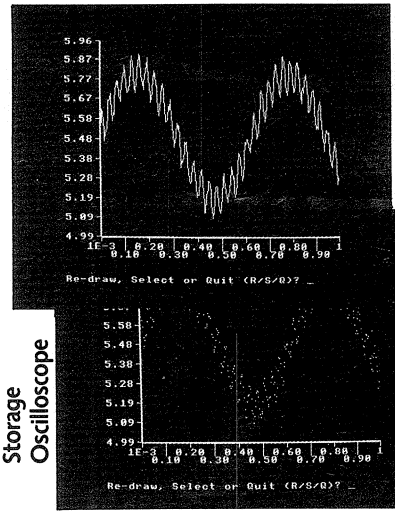
REVIEWS

REGULAR ITEMS

HINTS AND TIPS

GENERAL

MASTER



Storage Oscilloscope

Basic Program Comparator

Filer Accounts

Teletext ROM

EDITORIAL JOTTINGS

BEEBUG

In the past month we have implemented a number of substantial changes in the organisation of BEEBUG, changes which particularly affect subscriptions and orders. In order to explain fully what is happening and the reasons, we have devoted the whole of the first page overleaf to this important announcement. As a result we believe we shall be able to improve the service that we offer, and simplify the points of contact with BEEBUG for members.

MASTER COMPACT

At the PCW Show on the 3rd September, Acorn launched another member of the Master Series of micro-computers, the Master Compact. What is interesting to Acorn watchers is the manner in which the new machine is being marketed. This suggests that Acorn are taking a leaf from the Amstrad book, as the Compact will be sold as a complete packaged system (including 3.5" disc drive) and, initially, through high street multiples.

Acorn have most certainly undertaken their own market research and will clearly be hoping that they have identified a separate and successful market for the Compact. Some will undoubtedly see this as a replacement for the Electron, and to some extent that may be the case. The Compact, though, is streets ahead of the Electron technically. Indeed, standard benchmark tests show the Compact to be even faster than the Master 128.

The Compact is a most attractive system, with its own innovations. To many potential purchasers, in an increasingly applications oriented market, it may well seem a more desirable buy than the Master 128 itself. We only hope that Acorn have done their sums and produced the right machine at the right price. Only time will tell, but we certainly wish it every success.

BEEBUG SUPPORT

BEEBUG will be fully supporting the Compact as part of the Master Series, and all programs will be tested on this system. Likewise review products, where appropriate, will indicate compatibility with the new system. For this month, a letter 'C' will indicate compatibility with the Compact.

PROGRAM CLASSIFICATION

All programs in the magazine, and on the magazine cassette/disc, are marked with the symbols shown below. An uncrossed symbol indicates full working, a single line through a symbol shows partial working for that configuration (normally some modifications will be required), and a cross through a symbol indicates a program that will not work on that type of system. Similar symbols mark reviews where appropriate, but a single letter 'B' indicates compatibility with the model B (Basic I and II).

Basic I	I	Electron	
Basic II	II	Disc	
Tube		Cassette	
Model B+		Master 128	M
Compact	C	Model B	B

ANNOUNCEMENTS

Communication, Mailing and Distribution

Up to now we have been using a mailing agency in High Wycombe to handle your subscriptions and BEEBUGSOFT orders. We have now made the decision to take all of this in-house, and hope to be able to offer a better service as a result. From now on you simply use one address for ALL correspondence, magazine subscriptions, queries, mail order etc. That address is:

BEEBUG Ltd,
Dolphin Place,
Holywell Hill,
St Albans,
Herts AL1 1EX.

Please help us by indicating clearly to which section of BEEBUG your letter is addressed (e.g. editorial, mail order, technical support etc).

This is also our registered office address and we would ask all account holders, and suppliers, to note this address on their files.

Our multi-line switchboard is on 0727 40303, and technical enquiries are on 0727 60263 (10am - 4pm).

Additional Membership Benefits

MEMBERS DISCOUNTS

We are delighted to be able to offer members a 5% discount on all products available in our shop or through mail order. We already offer members a 25% discount on all BEEBUGSOFT products and this will remain in force. The enclosed Retail/Mail Order price list now reflects this new discount scheme by quoting both members and non-members prices. If you order from us it is ESSENTIAL that you quote your membership number (found on your address label) in order to be able to take advantage of the discounted prices.

TECHNICAL SUPPORT

We have re-organised our technical support team: Robert Barnes, Saleem Agowun and Peter Dunn (available on 0727 60263 between 10am and 4pm), are able to offer enhanced support encompassing: Information on the range of products supplied in our Retail/Mail Order department, BEEBUGSOFT software, all Acorn and other third party hardware and software, Technical support, in fact anything that you think we should be able to help you with. Our lines are bound to be busy, but keep trying. We will respond to written queries using standard broadsheets wherever possible. More complex matters will be dealt with separately, but may take a little longer.

MEMBERSHIP CARDS

We have produced a credit card sized BEEBUG membership card in elegant red and black. This will automatically be sent to new members and existing members as they renew their subscription. You may renew your subscription at any time to receive the card.

FREE PERSONAL ADVERTISEMENTS

Last month we introduced FREE personal advertisements for BEEBUG members. This has already proved exceedingly popular. Don't forget that you may place "WANTED" advertisements too. We may have to limit the number of ads that appear each month, but will place the advert in the next available issue. Any ads left over will be carried forward to the next issue unless you tell us otherwise. Please note that adverts should be sent in by 20th of each month to have a good chance of appearing in the next issue (usually published 4 weeks later). We will also continue to accept your business ads at the same rates as before (see supplement).

TRADE IN YOUR BBC FOR A MASTER OR COMPACT

Many members would like to be able to buy a Master 128 or Master Compact, but cannot quite justify the cost without some return from selling their BBC Micro. We now offer a trade-in service. Trade-in prices can be found in this month's supplement.

Members may also buy the used, refurbished, fully tested and guaranteed machines. This is proving to be very popular with educational establishments many of whom are members of BEEBUG.

BEEBUG SOFT FORUM

Converting Commsoft Pack Screens and Teletext

Both the Commsoft ROM, and the Beebugsoft Teletext Editor are capable of producing and editing mode 7 screens, but it is not possible to load ordinary mode 7 screens into the Commsoft editor for further editing. Moreover, screens produced by both products cannot be loaded into programs expecting a normal mode 7 load address of &7C00.

The short program listed below will change all this. And it may also be used to shift screens up or down by any number of lines; very useful if you have created some screens for Prestel or a Prestel simulator, but have forgotten to leave the top line clear for the Prestel header.

When run it requests two filenames: one for the screen to be converted, and the other to be used for the converted copy. These may be the same, in which case the new screen will overwrite the old. It then asks for two addresses to be given in hex. These are the load and save addresses. The program displays a table of addresses to be used for the various conversions envisaged. If you wish to shift a screen up or down, use a save address of &7C00, and a load address as follows:

Shift down		Shift up	
7C28	1	7BD8	1
7C50	2	7BB0	2
7C78	3	7B88	3

Once the selections have been made, the screen is loaded in for inspection purposes. If all is well, a press of the space bar will re-save it with the new parameters. Pressing Escape will abort the process.

```
  10 REM SCREEN ADDRESS
CONVERTER
  20 MODE7:HIMEM=&7000
  30 REPEAT
  40 CLS
  50 PRINT'"SCREEN ADDRE
SS CONVERTER"''
  60 PRINT"Mode 7 to Com
msoft - 7C00 7B98"
  70 PRINT"Commsoft to M
ode 7 - 7B98 7C00"
  80 PRINT"Teletext to M
ode 7 - 7C00 7C00"
  90 PRINT"Mode 7 to Tel
etext - no change"
 100 INPUT'"Screen filen
ame "file$
 110 INPUT"New filename
"new$
 120 INPUT"Load address
&"load$
 130 INPUT"Save address
&"save$
 140 CLS
 150 PROCoscli("LOAD "+f
ile$+" "+load$)
 160 A=GET
 170 PROCoscli("SAVE "+n
ew$+" "+save$+" +400")
 180 UNTIL FALSE
 190 :
 200 DEFPROCoscli(word$)
 210 $&380=word$
 220 X%=&80:Y%=3
 230 A=USR(&FFF7)
 240 ENDPROC
```

ADFS Masterfile II

Beebugsoft's Masterfile II has been re-written to work using the ADFS (Advanced Disc Filing System). ADFS Masterfile II is a completely separate version and in order to take full advantage of the ADFS will not work using the standard DFS. However, any files already created using the standard DFS can easily be copied using the Utility Disc supplied with the ADFS itself.

ADFS Masterfile II will work on the Master 128 or Master Compact, or with any BBC Micro fitted with the ADFS.

The programs take full advantage of the increased capacity of the discs - for example, whereas using the standard DFS, a file on a double-sided 80-track disc was limited to 197k (about 1,900 typical name & address records), using the ADFS will provide 650k (over 6000 typical name & address records).

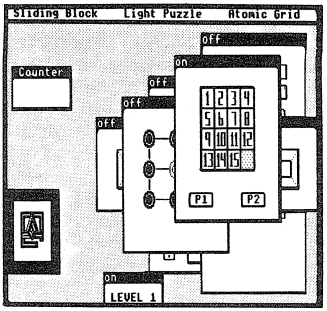
To swap your DFS version for the ADFS version simply return your ORIGINAL disc (in a suitable disc mailer) plus £5. Please do NOT return the entire pack.

ADFS Hardware Upgrade

To upgrade your BBC Micro to the ADFS you will first need to fit the 1770 disc controller. This is a straightforward operation for most machines, and can be done at home or at our showroom should you wish. The ADFS is supplied on ROM.

The 1770 disc controller, ADFS and ADFS Masterfile II are on special offer this month.

News News News News News News Ne



When the Cat's Away

It had to come: AMS has released a series of mouse-driven games for use with the popular AMX mouse. Two discs are available. Mind Games costs £14.95 and contains five games - Sliding Block Puzzle, Light Puzzle, Atomic Grid, Code Breaker, and Repeater. The second disc has one arcade game called Zap Zone for £4.95. Further details from AMS on 061-483 2737.

More Colours

Two add-ons for the BBC micro extend the graphics capabilities of the machine. Both devices allow the Beeb's normal selection of colours to be redefined from a palette of a wide range of alternative shades. The Plus card from CTS Recognition gives users the choice of eight colours in mode 2 from a palette of 4096 hues. Plus costs £99 from CTS on 0273-420897. Wild Vision's Palettemate board allows 16 colours to be selected from 4096 in mode 2. Palettemate costs £146.63. Further details on 091-281 7861.

A Better Box

BML Electronics' Barry Box (reviewed in BEEBUG Vol.5 No.1) has been improved to include several additional features. An auxiliary input to accept signals from HiFi and musical instruments is now built in. A 'tuning fork' facility allows the Barry Box to generate frequencies with an accuracy better than 0.1 Hz. A frequency recognition facility will calculate the frequency of a sampled sound (even if it has a large number of harmonics). The price of the improved Barry Box remains the same and upgrades are available to existing users. Further details from BML on 0908-640805.

Free Modems

To encourage more BBC owners to join Micronet free modems are offered to anyone taking out a full year's subscription to Prestel and Micronet. The free modem for the BBC micro is the Modem 2000. Alternatively, a free Micronet subscription (3 months) is available to purchasers of BEEBUG's sophisticated Magic Modem. Details of both offers are included with this issue. Micronet is on 01-278 3143.

New CAD

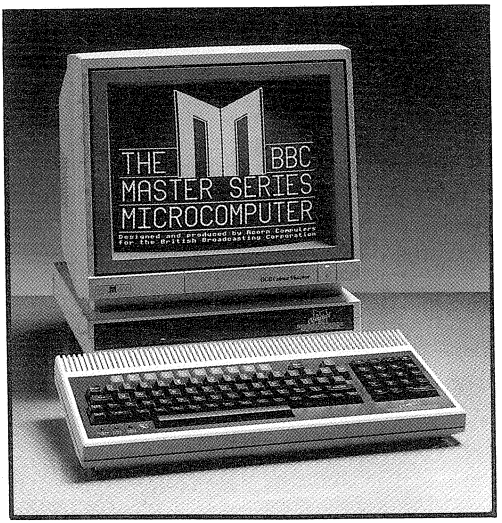
A low cost CAD system for the Beeb is available from Edusoft. Educad may be used for the construction of geometric and schematic diagrams. It includes a zoom facility for detailed work, a font editor for customised labelling and screen dump and plotter driver routines. Geometric accuracy is maintained with the use of extended 32 bit floating point mathematical functions. Educad costs £92. Details from Edusoft on 05436-76939.

Ultrafree

BBC Soft is giving away software. A set of new utilities to accompany the Ultracalc 2 spreadsheet ROM is available to users in exchange for a blank disc. The utilities cover a business graph and chart generator, a utility to transfer Ultracalc files to other spreadsheets and to Acornsoft's 'Database', and others. Interested Ultra-calc owners should send a blank formatted disc along with the return postage to BBC Soft, 35 Marylebone High Street, W1M 4AA.

Trivial News

Now that Trivial Pursuit has sold nearly 70 million copies worldwide and 3 million in this country it is inevitable that someone should come up with a computer version. The someone is Domark who claims that the computer version stays faithful to the original. It not only includes many of the original questions but new ones have been added to take advantage of the sound and graphics available. Trivial Pursuit costs £14.95 from Domark on 01-947 5622.



THE MASTER COMPACT

The Master Compact is Acorn's new machine for the highly competitive home market. Acorn have also taken a leaf out of Amstrad's book by selling a complete packaged system. Mike Williams reports on Acorn's newest, and maybe best machine in the Master series.

The recent launch of the Master Compact marks a new approach for Acorn, though one that has been adopted with notable success by rivals Amstrad. For the first time, Acorn are offering a complete ready-to-run system.

The new Compact will initially be aimed at the home computer market through high street retailers, with more emphasis on the educational (particularly primary) and small business markets in the new year. An 'Olivetti' version will be marketed in Italy at the same time, and other 'national' versions may follow.

System One:	128K RAM, 64K ROM 3.5" disc unit and power supply.
Price:	£399 + VAT
System Two:	As System One but with high-res monochrome monitor.
Price:	£469 + VAT
System Three:	As System Two but with medium-res colour monitor in place of monochrome version.
Price:	£ 599 + VAT

GENERAL DESCRIPTION

As the name suggests, the new machine is part of the Master Series, first seen in January, though it is distinctively different in several ways. The complete 'three-box' system comprises a keyboard containing the micro, a substantial 'power-box' with built-in 3.5" disc drive, and a choice of either a 'high-res' monochrome (green) screen, or a 'medium-res' RGB colour monitor. The power-box acts as a base for any monitor as in the illustration. The usual Welcome Guide and disc complete the package.

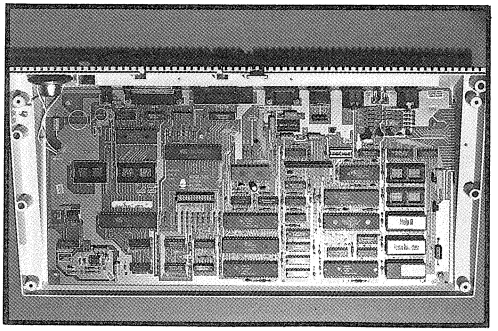
The general appearance is smart and clean, with none of the boxes and wires so commonly associated with the model B or Master 128. The keyboard layout, with one exception, is identical to that of the Master 128, including the separate numeric keypad. The keyboard has a somewhat spongy feel to it, and is slightly 'dished' from top to bottom in the style supposedly preferred by typists. The keyboard box is a little deeper than the key area and slopes upwards from front to back. The one difference is the new 'CODE' key in the place of the @ key (now moved above the zero). This can be used to generate the additional 128 characters of the Master Series which were not previous accessible from the keyboard. These include Greek characters, accented characters and other special symbols.

The power-box, with its built-in disc drive, is of very solid metal and plastic construction to provide a firm base for any monitor. This is plugged into the mains supply and a low voltage lead connected to the keyboard, with a ribbon cable for the disc connection. The power-box provides a power socket for the monitor, which in turn is connected to the Compact's video or RGB socket.

The whole system is most attractive and the cables are quite unobtrusive, though the disc ribbon-cable connection is somewhat unwieldy. The height of the power-box, only just above the back of the keyboard, means a gap must be left between the two units to give access to the disc drive. A second 3.5" disc drive can be fitted into the same unit as the first, or a 5.25" floppy drive connected externally. Such additions are likely to be provided by third party dealers rather than Acorn.

SYSTEM CONFIGURATION

The Master Compact is essentially a cut-down Master 128. It is provided with 128K of RAM of which 20K is shadow RAM, 12K private RAM, and 64K sideways RAM, as for the Master Series generally. One 64K ROM is supplied, and there are four empty ROM sockets, three for 16K ROMs and one for either 16K or 32K ROMs. The machine contains a disc interface, a printer port, black/white video and colour RGB connectors, an Atari-style joystick (or mouse) port, and an expansion port. This is similar but not identical to the Master 128 cartridge socket (logically), and can support sideways ROMs 0 and 1. Both the disc interface and the printer port use connectors different from those used previously by Acorn.



There is no cassette, Tube, 1MHz bus or user port (though some lines of the latter are available through the new digital joystick port and the expansion port), and the Analogue to Digital port has gone too. The serial port connector is still present, but the interface is now an optional extra (four chips) and to RS232 not RS423 specification.

The built in ROM software comprises the MOS (Machine Operating System) version 5.0, Basic VI (very similar to Basic IV in the Master 128), and the ADFS (Advanced Disc Filing System). No standard DFS is provided, and Acorn have clearly decided that the potential and extra storage capacity of this format more than outweighs its complexity. In addition, the Welcome disc contains a wealth of extra software, in particular View (Acorn's wordprocessor), TimPaint (an icon-style drawing package), Logotron's version of Logo, ABC (an award-winning educational word processing package for younger children), and Desk Top (a windows and icons style package providing calculator, card index, note pad and clock).

No doubt Acorn have analysed the market for the Compact, but particularly as most of the software is on disc, and in the form of ROM images for loading into sideways RAM, it is perhaps a shame that some thought wasn't given to providing more than one flavour of Welcome package. For starters, there could have been a more strongly educational version, one for the small business user (with spreadsheet and database), as well as one for the home user. It does seem an omission not to include ViewSheet (as with the Master 128), though it will be provided with the Olivetti version.

The Welcome Guide for the Master Compact is similar in style to that for the Master 128, and once again is likely to be subject to some criticsm. There are many facets of the new machine, and software on the Welcome disc, that receive scant if any mention in the Welcome Guide. No doubt users will be obliged to buy further reference manuals at comparatively high prices. More importantly, I also believe that the Welcome Guide fails in what must be its more important objective, introducing the new computer user to the world of micro-computing and the Acorn Compact in particular.

Of course this is a difficult task made even more difficult when such software as the ADFS is included, but the effort would, I am sure, have been worthwhile. I am also not keen on the page numbering - each section has a letter and is then numbered from 1 - as it is difficult to judge where any given page will fall. There are also a number of errors and inconsistencies, some of which,

I am assured, will be rectified in future editions.

THE WELCOME DISC

As more and more computer users are viewing the computer as a tool, rather than as a programming machine, I will concentrate on the software contained on the Welcome disc for the remainder of this review.

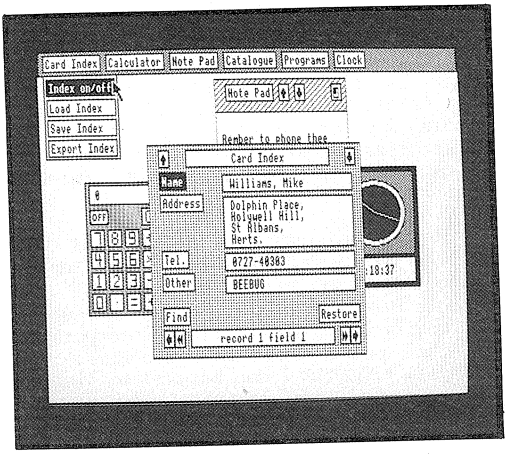
Inserting the Welcome disc and pressing Shift-Break displays the Master logo, soon followed by a menu-bar across the top of the screen, an arrow pointer and a window with instructions. The menu choices available are:

Graphics
Tutorials
Applications
Games
Utilities
Screen Colour

Moving the pointer to any one of these, and pressing Return results in a pull-down menu giving further choices. The graphics demos are the same as those for the Master 128, good for showing the newcomer what can be achieved, while the tutorials provide a reasonable introduction to certain aspects including the ADFS. There are two games, both no more than adequate, but acceptable in the circumstances. The Utilities include a set of useful ADFS disc utilities for more advanced ADFS users (quite undocumented in the Welcome Guide). Several other utilities are provided, including a pattern editor and a character definer. This allows alternative fonts to be designed, and several such are included on the Welcome disc.

Although the Compact has no battery-backed RAM, it is still able to save a start-up configuration using a device called an EEPROM - an EPROM that can be electrically written to. The utilities include a Control Panel, re-designed and much easier to use than that supplied with the Master 128, and this can be used to customise and save your own start-up configuration. Acorn advise that you cannot write to the EEPROM more than 10000 times, but this should be more than enough for most people and provides a neat, alternative solution to the battery-backed RAM of the Master 128. Indeed, you can also save less frequently used configurations on disc and load them from there, and load the standard default configuration in this way too.

By far the most interesting selection of software is found under the heading



Applications, and the most interesting entry of all is Desk Top. This provides a Mac look-alike WIMPS environment - WIMPS standing for Windows Icons Mouse Pointer System. A new menu-bar now offers Card Index, Calculator, Note Pad, Catalogue, Clock and a return to the Welcome menu. Selecting any of these results in a pictorial representation of the selected object being displayed on the screen. In addition, each object can be repositioned on the screen, and the size of the object can be changed. Unfortunately, there is no facility for saving your customised 'desk top', and you are forced to load the standard version first each time.

Desk Top provides a good demonstration of what can be achieved, but operation is rather sluggish, and in some cases too cumbersome for serious use - the Calculator is a case in point, though this can be operated from the keyboard rather than using the pointer to 'press' the buttons on the calculator. The Card Index provides a 125-record name and address file, and Acorn have thoughtfully provided such a file with their own name and address as the first record. This was quite interesting to experiment with, but limited for serious use. The Note Pad has 16 pages, and as with Card Index, several versions may be saved to disc and recalled as required. Both Card Index and Note Pad

will allow their contents to be 'exported' to View, and a Card Index file can be used to create a mail-merge facility for standard letters.

Clock provides both a digital and analogue clock display, but as the Master Compact has no real-time clock or battery back-up, the time must be set when the clock is first selected after power-up. The Catalogue option provides a neat, simple way of browsing through ADFS directories, renaming, deleting or running files, and also offers a '*' command facility.

What is even more interesting is that all the icon software is written in Basic, using sprites to define many of the symbols used, and all these facilities can be modified by the user, or incorporated in the user's own WIMPS programs. For example, *POINTER 1 can be used to display an arrow pointer on the screen, and *POINTER Ø to remove it. Not only are all the graphics extensions of the Master 128 included in the Compact's MOS, but the sprites from the GXR Graphics Extension ROM for the model B are provided on disc. Indeed, according to Acorn, the sprites were originally developed to support the WIMPS-style environment now offered on the Compact. Unfortunately, none of this is documented in the Welcome Guide, and until Acorn release more advanced manuals for the Compact, users will have to explore the WIMPS software on the Welcome disc themselves.

For serious word processing, View 3 is supplied complete with five undocumented printer drivers on disc, in addition to the built in Epson printer driver. In a simpler vein, the ABC word processor is also available, and text created with this can subsequently be transferred to View.

For those who are more interested in Basic programming, the Compact not only has Basic VI in ROM, claimed to be even faster than the Basic IV of the Master 128 (as a result of rewritten floating point routines), but Hi-Basic called Bas128 can be loaded from disc, as can a ROM image of Basic IV, presumably provided for compatibility with the rest of the Master Series.

CONCLUSIONS

Personally, I think I would rather have a Master Compact than a Master 128, but then I like the simple uncluttered life. I would certainly have preferred that the serial interface (for communications) was included as standard, and if not the interface, then why not a built-in modem as well? However, that is really wanting to have your cake and eat it too, and I am sure that Acorn have fairly carefully targeted this machine pricewise.

The software bundled in with the Compact is good, but could have been better, particularly if ViewSheet had been included as well as View. My main criticism concerns the Welcome Guide which I find both too technical for the beginner and yet too limited for the experienced user. And with such a wealth of software included it does seem a shame not to make more of this.

Acorn have high hopes for their Compact in the pre-Christmas retail market and it certainly deserves to do well. No doubt the price of this latest Acorn product will come under attack, but it is a quality product (as always from Acorn). I would have preferred to see the new machine retailing at under £500 including VAT, and I hope that Acorn have got their sums right by producing a machine at a price that people will want to pay, rather than at the level that Acorn believe it should be. Despite that, anyone seeking a good home computer for serious use would do well to consider the Compact.

With so much software to investigate, we shall certainly be returning to Acorn's new Compact in future issues of BEEBUG.

STOP PRESS

Acorn announced on the 28th August that because of difficulties in obtaining TV modulators they would be offering two alternative versions of System One, one with TV modulator at £395 + VAT, and one without at a price of £385 + VAT.

Prices were also announced for the RS232 upgrade at £26, additional 3.5" disc drive kit at £99, and PAL TV adaptor at £26 (all prices + VAT).

The complete View family of products has also been released for the Compact.

WHICH MACHINE?

Is it an Electron, a model B or even a Master Compact? Does it have a second processor or printer connected? Is the DFS or ADFS currently active? David Graham gives some routines, to add to your own programs, that provide all the answers.

When writing software of any kind that may be used on more than one machine configuration it is very useful to include in that software, routines that will check the status of the host computer system, and react accordingly. You may, for example, require a program to test for the presence of the Tube, and inform the user to turn it off. Or you may have a program which was written on a BBC B, but which needs small modifications to run on a Master, or on an Electron.

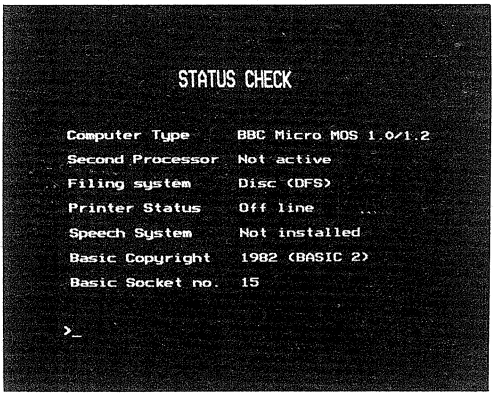
The program presented here contains a battery of tests which may be used for this purpose, either together or in isolation. First of all type it in and run it. It will work on any Acorn clone, and will display the following details:

Machine type and operating system
Tube status
Current filing system
Printer status
Speech system status
Basic version and socket number
(providing that the Tube is inactive)

Each test uses a separate function so as to enable individual routines to be easily incorporated into other programs. The only major proviso to this is that the test for the Basic version number will not work across the Tube. In the present program the Basic test is therefore only performed if the Tube test is negative. In the rest of this article we will take a closer look at some of the routines used.

MACHINE TYPE

The use of negative INKEY gives a comprehensive machine configuration and operating system test which is easily incorporated into any program. INKEY(-256) gives a unique response for the various Acorn clones. These are given in full in the table overleaf. To test your machine, simply type: PRINT INKEY(-256)



FILING SYSTEM

Another important feature to test for is the currently active filing system. You may well require a program to behave differently if it is using cassette rather than disc for example. This can be accomplished through the call OSARGS, which will even distinguish between DFS and ADFS. After executing the line:

```
A%=0:Y%=0:Z%=USR(&FFDA) AND &FF
```

the variable Z% will contain the currently active filing system number as follows:

Z%	FILING SYSTEM	Z%	FILING SYSTEM
0	None	5	Econet
1	Cassette (1200)	6	Telesoft (TFS)
2	Cassette (300)	7	IEEE
3	ROM	8	Disc (ADFS)
4	Disc (DFS)	12	RAM (RAFS)

This test is incorporated into the function FNfilingtest.

TESTING BASIC

The Basic test is handled by FNbasictest. This looks a little convoluted because it has to make two operating system calls. The first, to OSBYTE 187, establishes the socket number of Basic. Then OSRDSC is used to read the last digit of the copyright message of the Basic ROM. This corresponds to the Basic version number.

```
10 REM Program STATUS TESTER
20 REM Version B0.6
30 REM AUTHOR  David Graham
40 REM BEEBUG  October 1986
```

```
  50 REM Program subject to copyright
  60 :
 100 MODE7
 110 PRINTTAB(11,2)CHR$134;CHR$141;"STA
TUS CHECK"
 120 PRINTTAB(11,3)CHR$134;CHR$141;"STA
TUS CHECK"
 130 :
 140 REM TEST FOR MACHINE TYPE
 150 PRINTTAB(0,7)CHR$131;"Computer Typ
e    ";CHR$134;
 160 RESTORE 740
 170 PRINTFNdisplay(INKEY(-256))
 180 :
 190 REM TUBE TEST
 200 PRINTTAB(0,9)CHR$131;"Second Proce
ssor ";CHR$134;
 210 IF FNtubetest PRINT"Active" ELSE P
RINT"Not active"
 220 :
 230 REM FILING SYSTEM TEST
 240 PRINTTAB(0,11)CHR$131;"Filing syst
em    ";CHR$134;
 250 RESTORE 860
 260 PRINTFNdisplay(FNfilingtest)
 270 :
```

INKEY(-256)	OPERATING SYSTEM
-1	BBC MOS 1.0/1.2
0	BBC MOS 0.1
1	Acorn Electron MOS 1.0
245	Master Compact MOS 5.0
250	Acorn ABC MOS
251	BBC B+ MOS 2.0
252	BBC Germany
253	BBC Master (UK) MOS 3.0
254	BBC US MOS 1.0 or 1.1

	BASIC VERSIONS
I	1981 Standard BBC
II	1982 Enhanced BBC, B+ and Electron
III	1983 Hi Basic (External 2nd Processor)
IV	1984 Master Series
V	1985 Hi Basic (Internal Co-processor)
VI	1986 Master Compact

```
 280 REM PRINTER TEST
 290 PRINTTAB(0,13)CHR$131;"Printer Sta
tus    ";CHR$134;
 300 IF FNprintertest PRINT"On line" EL
SE PRINT"Off line"
 310 :
 320 :REM SPEECH SYSTEM TEST
 330 PRINTTAB(0,15)CHR$131;"Speech Syst
em    ";CHR$134;
 340 IF FNspeechtest PRINT"Operational"
 ELSE PRINT"Not installed"
 350 :
 360 REM BASIC (I/O PROCESSOR) TEST
 370 IF FNtubetest THEN PRINT'':END
 380 B$=FNbasictest
 390 PRINTTAB(0,17)CHR$131;"Basic Copyr
ight  ";CHR$134"198";B$;
 400 PRINT" (BASIC ";B$;")"
 410 PRINTTAB(0,19)CHR$131;"Basic Socke
t no. ";CHR$134;Y%
 420 PRINT''
 430 END
 440 :
 450 DEFFNdisplay(Z%)
 460 REPEAT
 470 READ W%,W$
 480 UNTIL W%=Z% OR W%=1000
 490 =W$
 500 :
 510 DEFFNtubetest
 520 A%=&EA:X%=0:Y%=&FF
 530 =USR(&FFF4) AND &FF00
 540 :
 550 DEFFNfilingtest
 560 A%=0:Y%=0
 570 =USR(&FFDA) AND &FF
 580 :
 590 DEFFNprintertest
 600 VDU2,1,0,1,0,1,0,1,0,3
 610 =(ADVAL(-4)=63)
 620 :
 630 DEFFNspeechtest
 640 A%=&EB:X%=0:Y%=&FF
 650 =USR(&FFF4) AND &FF00
 670 DEFFNbasictest
 680 X%=0:Y%=&FF:A%=&BB
 690 Y%=(USR(&FFF4) AND &FF00) DIV &100
 700 ?&F6=&15:?&F7=&80
 710 =CHR$(USR(&FFB9) AND &FF)
 720 :
 730 REM DATA FOR MACHINE TYPE
 740 DATA-1,BBC Micro MOS 1.0/1.2
 750 DATA0,BBC Micro MOS 0.1
 760 DATA1,Acorn Electron
 770 DATA245,Compact MOS 5.0
 780 DATA250,Acorn ABC Machine
 790 DATA251,BBC B+ MOS 2.0
 800 DATA252,BBC Micro (Germany)
 810 DATA253,Master Series (UK)
 820 DATA254,BBC Micro (USA)
 830 DATA1000,Unrecognised system
 840 :
 850 REM DATA FOR FILING SYSTEM
 860 DATA0,None
 870 DATA1,Cassette 1200 baud
 880 DATA2,Cassette 300 baud
 890 DATA3,ROM filing system
 900 DATA4,Disc (DFS)
 910 DATA5,Econet
 920 DATA6,Telesoftware
 930 DATA7,IEEE filing system
 940 DATA8,Disc (ADFS)
 950 DATA12,RAM filing system
 960 DATA1000,Not recognised
```

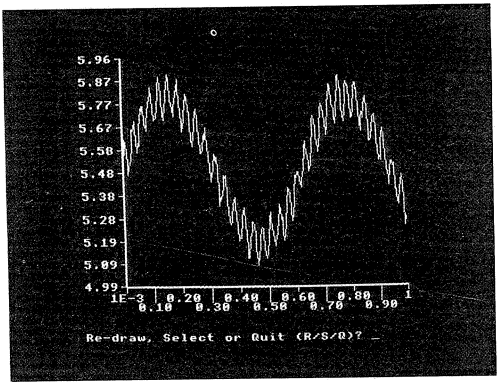
DIGITAL STORAGE OSCILLOSCOPE (Part 1)

David Peckett shows how to turn your micro into an oscilloscope. His program samples and displays signals appearing at the analogue port.

With its combination of high definition graphics and analogue input port, a standard Beeb can easily form the basis for an effective digital storage oscilloscope. Its chief limitation is its frequency reponse. It can only handle input frequencies up to around 50Hz. This is determined by the maximum sample rate of the Beeb's analogue to digital converter (ADC), but is nevertheless more than sufficient for many interesting applications. For example, a simple diode temperature sensor would allow it to plot a temperature curve over any chosen period, or it could be used to monitor noise levels, or even plot the response of an amplifier or a filter given a suitable sweep generator.

First though, what is a storage oscilloscope? It's an instrument for displaying electrical waveforms which does not just show the data as it happens. It stores the signal, and allows you to examine and measure all or part of it at your leisure. The present program allows you to 'build' one of these devices without touching a soldering iron or a single resistor; though you will need to plug a suitable device into analogue port 1 to produce meaningful results.

In fact, the oscilloscope program is supplied in two parts. Part 1 (this month) will work on its own to both record and display data. Next month we will be adding further routines which implement star commands, loading and saving of data to or from disc or cassette, and provide for flexible scaling of the time axis of the display. If any of these options are selected from the current program they will be effectively ignored. What actually



happens is that one of four empty procedures is called. These appear at the end of the program, and will be replaced next month with the full routines.

USING THE PROGRAM

When you first run the program, a menu gives the choice of recording data, loading it from an existing file, using a star command, or quitting. Since the load and star command options will not be implemented until part 2, we will assume that you wish to record. Pressing "R" will take you into a dialogue to define the recording parameters. You may choose the time between samples (anything from 0.01secs to 10^6secs!), and the number of samples to be taken (max 2000). You are also told how long the sampling process will take, with a chance to change your mind, just in case you didn't really mean to take 2000 samples at 1 hr intervals.

Having set up the system, press the space bar to start recording. If the time between data points is 0.02 secs or more, the incoming signal will be displayed directly on the screen as it is recorded. At faster speeds the display is suppressed as the BBC B is unable to keep up, though a Master or Master Turbo could probably cope. Press the space bar again at any time if you want to stop recording early.

When the recording is over, a further press of the space bar takes you back to the main menu, to which has now been added options to plot the data and to save it (though save is not implemented yet). Selecting plotting, by pressing "P", takes you to the next stage of the program.

First, you must scale the data. Data from the ADC always has an integer value between 0 and 65535, corresponding to input voltages in the range 0 to 1.8 volts. When you use the 'scope, the chances are that you will use a transducer (e.g. a potentiometer) to convert whatever you are actually measuring (angle, temperature, pressure, etc) to a voltage suitable for the ADC.

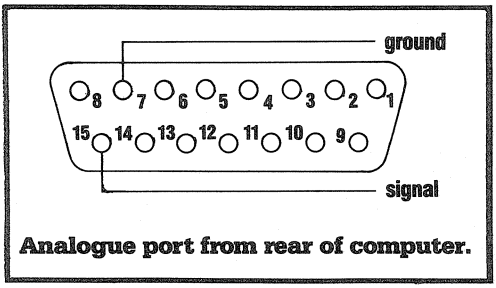
The program asks you for the extreme limits of the measured input - the 0V and 1.8V levels - so that it can display a sensible Y-axis scale. So, if you are measuring temperature, and -10 degs corresponds to a zero input to the ADC, and 100 degs means full range, answer "-10" and "100" to the program's questions. The X-axis is always scaled with the sample size. You can then choose to display the data as a continuous line, or on a point-by-point basis.

Finally, the program plots the data on the screen. It sets the axes with a simplified version of the graphing routine from the excellent Workshop article in the Beebug Vol.5 No.8. To fit enough data on the screen, there is only room for scales to be 4 characters wide. To get around this the scales use a form of engineering notation (e.g. milli and mega) and you must read them carefully. For more details, I refer you to the Workshop article itself. While on the subject of available space, it is worth mentioning that users of shadow RAM can obtain a higher resolution display by using mode 0 in place of mode 4. If you have a Master, simply replace line 110 with:

```
110 MODE 128
```

Once you have displayed the data, you can either re-plot it or "Q"uit to the main menu. The "S" option, which will be implemented in part 2, will allow you to select any part of the graph for closer examination.

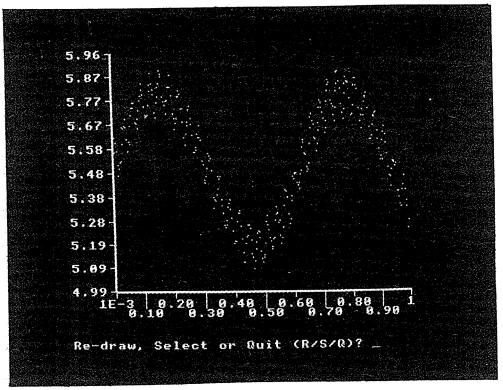
Next month we will supply the remaining routines and discuss the program itself a little more fully, as well as giving details of how to plot mathematical functions.



Analogue port from rear of computer.

```
  10 REM Program Digital Scope
  20 REM Version B4.0
  30 REM Author  David Peckett
  40 REM Beebug  October 1986
  50 REM Program Subject to Copyright
  60 :
 100 ON ERROR GOTO 2920
 110 MODE4:PROCInit
 120 ON ERROR GOTO 2920
 130 REPEAT
 140 PROCMenu
 150 IF Opt$="*" THEN PROCStar
 160 IF Opt$="O" THEN PROCOld
 170 IF Opt$="P" THEN PROCPlot
 180 IF Opt$="R" THEN PROCRecord
 190 IF Opt$="S" THEN PROCSave
 200 UNTIL Opt$="Q"
 210 CLS
 220 END
 230 :
1000 DEF PROCInit
1010 Maxpts=2000
1020 DIM Data%(Maxpts),osclibuf 30
1030 N%=0:P%=1:*FX16,1
1040 ENDPROC
1050 :
1060 DEF PROCMenu
1070 CLS:COLOUR 129:COLOUR 0
1080 PRINTTAB(8,2) " DIGITAL OSCILLOSCO
PE "
1090 PRINTTAB(16,4)"OPTIONS"
1100 PRINTTAB(10,8)"R"TAB(10,15)"O"TAB(
10,17)"*"TAB(10,20)"Q"
1110 IF N% THEN PRINTTAB(10,10)"P"TAB(1
0,13)"S"
1120 COLOUR 128:COLOUR 1
1130 PRINTTAB(11,8)"ecord Data"
1140 PRINTTAB(11,15)"ld Data"
1150 PRINTTAB(11,17)" Star Command"
1160 PRINTTAB(11,20)"uit the Program"
1170 IF N% THEN PRINTTAB(11,10)"lot the
 data"TAB(12,11)"(";N%;" points availabl
e)"TAB(11,13)"ave Data"
1180 PRINTTAB(10,24)"Which option? ";
1190 IF N% THEN Opt$=FNGetch("*OPQRS")
ELSE Opt$=FNGetch("*OQR")
1200 ENDPROC
1210 :
1220 DEF PROCRecord
1230 hival=-1
1240 REPEAT
```

```
 1250 CLS:COLOUR 129:COLOUR 0
 1260 PRINTTAB(14,3)"RECORD DATA"
 1270 COLOUR 128:COLOUR 1
 1280 REPEAT
 1290 PRINTTAB(5,8)SPC30
 1300 PRINTTAB(5,8)"No. data points (2-"
;Maxpts;")?";
 1310 INPUT " "S%
 1320 PRINTTAB(0,31)SPC35;
 1330 IF S%<2 OR S%>Maxpts THEN VDU7:PRI
NTTAB(0,31)"ONLY 2 - ";Maxpts;" ALLOWED"
;
 1340 UNTIL S%>1 AND S%<=Maxpts
 1350 N%=0
 1360 REPEAT
 1370 PRINTTAB(5,12)SPC35
 1380 INPUTTAB(5,12)"Sample time (secs -
.01 min)? "Tsamp
 1390 PRINTTAB(0,31)SPC35;
 1400 IF Tsamp<.01 THEN VDU7:PRINTTAB(0,
31)"MINIMUM TIME IS 0.01 SECS";
 1410 UNTIL Tsamp>=.01
 1420 PRINTTAB(5,16)"It will take ";(S%-
1)*Tsamp;" secs to read"TAB(8,17)"Is thi
s OK (Y/N)? ";
 1430 OK$=FNGetch("YN")
 1440 UNTIL OK$="Y"
 1450 T%=Tsamp*100+.5:U%=T%
 1460 PRINTTAB(5,24)"Ready - Press SPACE
 to start ";
 1470 REPEAT UNTIL INKEY-99
 1480 REPEAT UNTIL NOT INKEY-99
 1490 Q%=Tsamp>=.02
 1500 MOVE 0,200:PLOT 21,1279,200
 1510 IF Q% THEN VDU 24,0;204;1279;1023;
:CLG:MOVE 0,600
 1520 COLOUR 129:COLOUR 0
 1530 PRINTTAB(15,27)"RECORDING"
 1540 COLOUR 128:COLOUR 1
 1550 PRINTTAB(0,31)"Press SPACE to stop
 early ";
 1560 TIME=0
 1570 REPEAT
 1580 N%=N%+1:Data%(N%)=ADVAL(P%)
 1590 Data%(N%)=ADVAL(P%)
 1600 IF Q% DRAW N%,Data%(N%)/85+220
 1610 REPEAT UNTIL TIME>=U%
 1620 U%=U%+T%
 1630 UNTIL N%=S% OR INKEY-99
 1640 PRINTTAB(0,31)SPC35;
 1650 PRINTTAB(5,29)"FINISHED - ";N%;" p
oints recorded"
 1660 PRINTTAB(0,31)"Press SPACE to exit
 ";:SOUND1,-15,200,10
 1670 REPEAT UNTIL INKEY-99
 1680 VDU26
 1690 ENDPROC
 1700 :
 1710 DEF PROCPlot
```




```
 1720 CLS:COLOUR 129:COLOUR 0:PRINTTAB(1
5,3)"PLOT DATA":COLOUR 128:COLOUR 1
 1730 PROCLimits:first=1:last=N%
 1740 PRINTTAB(5,12)"Joined or Discrete
points (J/D)? ";
 1750 Line$=FNGetch("JD"):PRINT Line$
 1760 IF Line$="J" THEN LType=5 ELSE LTy
pe=69
 1770 PROCPlota
 1780 ENDPROC
 1790
 1800 DEF PROCPlota
 1810 LOCAL POpt$
 1820 PROCScale(first,last)
 1830 hival=H%*Scale+RWlo:loval=L%*Scale
+RWlo
 1840 PROCGraph(first,last,loval-(hival-
loval)*.1,hival+(hival-loval)*.1)
 1850 Xstep=FNMax((last-first)/250,1)
 1860 PROCDrawGraph
 1870 VDU 26:PRINTTAB(0,31) "Re-draw, Se
lect or Quit (R/S/Q)? ";
 1880 POpt$=FNGetch("RSQ")
 1890 IF POpt$="R" THEN PROCPlot:ENDPROC
 1900 IF POpt$="S" THEN PROCSelect:CLS:P
ROCPlota
 1910 ENDPROC
 1920 :
 1930 DEF PROCLimits
 1940 PROCScale(1,N%)
 1950 PRINTTAB(0,24)"Recorded data value
s from";L%;" to ";H%;" in a possible""
range of 0 to 65535"
 1960 INPUT"Real-world value equiv. to 0
? "RWlo
 1970 INPUT"Real-world value equiv. to 6
5535? "RWhi
 1980 Scale=(RWhi-RWlo)/65535
 1990 hival=H%*Scale+RWlo:loval=L%*Scale
+RWlo
 2000 PRINT"Data in range: ";loval'TAB(1
1)" to ";hival;
```

```
2010 ENDPROC
2020 :
2030 DEF PROCScale(f,l):LOCAL I%
2040 H%=-1:L%=99999
2050 FOR I%=f TO l
2060 D%=Data%(I%)
2070 IF D%>H% THEN H%=D%
2080 IF D%<L% THEN L%=D%
2090 NEXT
2100 ENDPROC
2110 :
2120 DEF PROCGraph(xlo,xhi,ylo,yhi)
2130 IF ylo>yhi THEN temp=ylo:ylo=yhi:y
hi=temp
2140 VDU 20,26:CLS
2150 xscale=1040/(xhi-xlo)
2160 yscale=800*Scale/(yhi-ylo)
2170 PROCaxes:PROCgrads:PROCwindows
2180 ENDPROC
2190 :
2200 DEF PROCaxes
2210 LOCAL X%,x1%
2220 MOVE 160,200:DRAW 1200,200
2230 MOVE 160,196:DRAW 1200,196
2240 MOVE 160,200:DRAW 160,1000
2250 MOVE 158,200:DRAW 158,1000
2260 MOVE 156,200:DRAW 156,1000
2270 FOR X%=0 TO 1200 STEP 104
2280 x1%=160+X%
2290 MOVE x1%,180-(X% MOD 208)*.32:DRAW
x1%,200
2300 NEXT
2310 FOR Y%=200 TO 1000 STEP 80
2320 MOVE 140,Y%:DRAW 160,Y%
2330 NEXT
2340 ENDPROC
2350 :
2360 DEF PROCgrads
2370 LOCAL sclfact,scl$,X%,Y%
2380 VDU 5
2390 sclfact=FNscale(xlo,xhi)
2400 FOR X%=0 TO 10
2410 scl$=FNval(xlo,xhi,X%)
2420 MOVE 160+X%*104-LEN(scl$)*16,172-(
X% MOD 2)*32
2430 PRINT scl$
2440 NEXT
2450 sclfact=FNscale(ylo,yhi)
2460 FOR Y%=0 TO 10
2470 MOVE 0,Y%*80+212
2480 PRINT RIGHT$(STRING$(4,CHR$32)+FNv
al(ylo,yhi,Y%),4)
2490 NEXT
2500 VDU 4
2510 ENDPROC
2520 :
2530 DEF FNscale(lo,hi)
2540 LOCAL temp,sf
2550 temp=ABS(hi)
2560 IF ABS(lo)>temp THEN temp=ABS(lo)
2570 sf=LOG(temp) DIV 3
2580 IF temp<1 THEN sf=sf-1
2590 =10^(sf*3)
2600 :
2610 DEF FNval(lo,hi,I%)
2620 LOCAL val,val$
2630 val=(lo+(hi-lo)*I%/10)/sclfact
2640 val$=LEFT$(STR$(val),4)
2650 IF RIGHT$(val$,1)="." THEN val$=LE
FT$(val$,LEN(val$)-1)
2660 =val$
2670 :
2680 DEF PROCwindows
2690 VDU 24,160;200;1279;1023;
2700 VDU 28,0,31,39,29
2710 IF yscale>0 THEN VDU 29,160;280-L%
*yscale; ELSE VDU 29,160;920-L%*yscale;
2720 ENDPROC
2730 :
2740 DEF PROCDrawGraph
2750 MOVE0,Data%(first)*yscale
2760 FOR X%=(first+1) TO last STEP Xste
p
2770 PLOT LType,(X%-first)*xscale,Data%
(X%)*yscale
2780 NEXT
2790 ENDPROC
2800 :
2810 DEF FNGetch(str$)
2820 LOCAL key$:*FX15,1
2830 REPEAT
2840 key$=GET$
2850 IF key$>="a" AND key$<="z" THEN ke
y$=CHR$(ASC(key$) AND &DF)
2860 UNTIL INSTR(str$,key$)
2870 =key$
2880 :
2890 DEF FNMin(a,b)=-(a<b)*a-(b<=a)*b
2900 DEF FNMax(a,b)=-(a>b)*a-(b>=a)*b
2910 :
2920 ON ERROR OFF
2930 VDU7:CLS:PRINT':REPORT:PRINT" at L
ine ";ERL
2940 PRINT'"Press Space to continue"
2950 PRINT"or Escape to exit":*FX15,1
2960 IF GET CLS:GOTO 120
2970 :
2980 DEF PROCStar:ENDPROC
2990 DEF PROCOld:ENDPROC
3000 DEF PROCSave:ENDPROC
3010 DEF PROCSelect:ENDPROC
```

This program will run on a Compact, but cannot, of course, record any data as the Compact has no analogue port. The program can, however, be used to load and display data captured previously on a model B or Master when part two is added.

Basic Program Comparator

Distinguishing between different versions of a program is easy with this utility from Jagdish Sah.

While developing programs I sometimes end up with several versions of the same program each with only minor differences. It is quite common to lose track of these differences in various files. To resolve this problem the utility listed here will compare two Basic programs line-by-line and list just those lines which do not match. One of the programs to be compared should be in memory at the current value of PAGE and the other one in a file on disc or cassette.

The utility can also be used to simply verify programs saved on tape or disc. When a program is saved, even on a disc, occasional spikes may lead to a corrupted copy. Most tape users may be familiar with the additional problem of difficulty in loading the saved program. After saving your program you can use this utility to check whether the saved copy matches exactly that in memory.

Type in the program as listed. Save it on tape or disc before you run it. If it does not produce any error message, save the machine code produced using the *SAVE command displayed. When you want to use it, type:

```
*RUN COMPRMC
```

This will implement a new command which has the syntax:

```
*COMPARE <fsp>
```

where <fsp> stands for a file specification. For example:

```
*COMPARE MYPROG
```

will compare the program, line by line, in memory (at PAGE) with that in the file MYPROG. The lines pertaining to the in-memory program will be preceded by M and those pertaining to the file by F. If the two files match exactly, then the message 'Files match' will be displayed.

The usefulness of the utility can be demonstrated with the help of this program itself. First make sure you have saved a copy of the typed program on tape or disc. If you want, you could verify whether the program was saved correctly by entering the command:

```
*COMPARE COMPARE
```

assuming that 'COMPARE' is the filename of the Basic program. If the message 'Files match' appear, this will assure you that the saved program is an exact image of that in the memory.

Next let us introduce some differences in the in-memory version deliberately to show how the utility behaves in such situations. Insert an extra space in one of the lines, say, in line 100. Also type in an extra line, say:

```
152 REM XYZ
```

Now use the *COMPARE command as described above. It will list the two pairs of lines which were altered. Since line 152 is not present in the file, a blank line will be shown against the prefix F.

The listing of the mismatched pairs is produced in paged mode. That is, after a screenful of listing you must press the Shift key to continue further. The Escape key may be pressed at any time to terminate the comparison process. If the specified file does not contain a Basic program, the utility will display "??" and will terminate immediately.

The program works under both tape and disc filing systems. Under the tape filing system the command may also be used as

```
*COMPARE ""
```

to compare the next program on the tape with that in the memory.

The new command will work in immediate mode only; i.e., it should not be included in a line of a Basic program. It will work under Basic I, Basic II, and Basic IV in the Master, but not under the Hi-Basic in the second processor.

PROGRAM NOTES

When a command is entered, the utility program intercepts the operating system's OSCLI routine to check whether it is *COMPARE. If so, it takes appropriate

action as described before.

One line at a time is read from the specified program file, and is stored at the TOP of the current program in memory. This line is then compared with the equivalent line in memory. The listing of the mismatched lines is produced with the help of routines in the Basic ROM itself. The ROM routines used are:

pchar	to print the character whose ASCII code is in the accumulator.
ptoken	to print the Basic keyword whose token is in the accumulator.
plineno	to print a line number.
plineno+4	to print a line number with the print field width of 5.
chkgoto	to convert line numbers after GOTO, GOSUB etc. These are present in a specially coded form in the program.
chkprog	to check whether a valid Basic program is present at PAGE. If not, a "Bad program" message is issued.
exit	to exit from this utility program and to return to the Basic interpreter.

The entry points for these routines for Basic I, II and IV are given in DATA statements at the end of the program. For the Compact load Basic IV from the Welcome disc first as this program will not work with the Compact's built-in ROM Basic.

When listing a line from the program file, it is recognised by the format &ØD, line number, line length and the Basic text. If the utility finds any departure from this, it displays "??" and terminates.

```
  10 REM PROGRAM COMPARE
  20 REM VERSION B0.2
  30 REM AUTHOR  Jagdish Sah
  40 REM BEEBUG  OCTOBER 86
  50 REM PROGRAM SUBJECT TO COPYRIGHT
  60 :
 100 start=&A00
 110 PROCassemble:PROCchecksum
 120 IF S%<>checksum%:PRINT'"Checksum e
rror"':END
 130 PRINT'''"Type"''"*SAVE COMPRMC ";~s
tart;" ";~P%
 140 PRINT'"to save the machine code pr
oduced."'
 150 END
 160 :
1000 DEF PROCassemble
1010 osbyte=&FFF4:osfind=&FFCE
1020 osbget=&FFD7:osasci=&FFE3
1030 osnewl=&FFE7:page=&18:top=&12
1040 lineptr=&B:lineptr2=&70:stack=&72
1050 lineno=&2A:lineno2=&73:handle=&75
1060 linelen=&76:linelen2=&77
1070 savex=&78:savey=&79:quoteflg=&7A
1080 pointer=&7B
1090 P=?&8015:IFP=49 THEN RESTORE 2600
ELSE IFP=50 THEN RESTORE 2630 ELSE IF P=
52 THEN RESTORE 2660
1100 READ pchar,ptoken,plineno,chkprog
1110 READ chkgoto,exit,checksum%
1120 FOR opt=0 TO 2 STEP 2
1130 P%=start:[OPT opt
1140 LDA &209:CMP #newrtn DIV 256
1150 BEQ out:STA oldrtn2+2:LDA &208
1160 STA oldrtn2+1:LDA #newrtn MOD 256
1170 SEI:STA &208:LDA #newrtn DIV 256
1180 STA &209:CLI
1190 .out RTS
1200 .oldrtn PLA:TAX:PLA:PLP
1210 .oldrtn2 JMP 0
1220 :
1230 .newrtn PHP:PHA:TXA:PHA
1240 LDX #0:STX lineptr:STX lineno+1:ST
X pointer
1250 .chkcmd LDA cmd,X:CMP &700,X
1260 BNE oldrtn:INX:CPX #9
1270 BNE chkcmd
1280 TXA:PHA:TSX:DEX:DEX:DEX
1290 STX stack:LDA #139:LDX #1
1300 LDY #0:JSR osbyte:JSR chkprog
1310 PLA:TAX:LDY #7:LDA #&40
1320 JSR osfind:TAX:BNE filefound
1330 BRK:OPT FNequb(&D6)
1340 OPT FNequs("File not found")
1350 OPT FNequb(0)
1360 .filefound STA handle:LDA page
1370 STA lineptr+1:LDA top
1380 STA lineptr2:LDA top+1
1390 STA lineptr2+1:LDA #117
1400 JSR osbyte:TXA:AND #1
1410 BNE mainloop:LDA #14:JSR osasci
1420 .mainloop LDY #0:JSR getabyte
1430 CMP #&0D:BEQ mloop2:LDA #ASC("?")
1440 JSR osasci:JSR osasci:JSR osnewl
1450 .goclose JMP close
1460 .mloop2 JSR getabyte
1470 STA lineno2+1:TAX:BPL mloop5
1480 AND lineno+1:BMI goclose
1490 BPL mloop3
1500 .mloop5 JSR getabyte:STA lineno2
1510 JSR getabyte:STA linelen2:TAX
1520 .getline JSR getabyte:DEX
1530 CPX #4:BNE getline
1540 .mloop3 LDY #3:LDA (lineptr),Y
1550 STA linelen:DEY:LDA (lineptr),Y
```

```
1560 STA lineno:DEY:LDA (lineptr),Y
1570 STA lineno+1:BPL mloop4
1580 AND lineno2+1:BMI close
1590 .mloop4 LDA lineno:CMP lineno2
1600 BNE difline:LDA lineno+1
1610 CMP lineno2+1:BNE difline
1620 LDA linelen:CMP linelen2
1630 BNE mismatch:LDY #4
1640 .cmpline LDA (lineptr),Y
1650 CMP (lineptr2),Y
1660 BNE mismatch:INY:CPY linelen
1670 BNE cmpline:BEQ incptr3
1680 .mismatch JSR listline
1690 JSR listline2
1700 .incptr3 JSR incptr:BCC mainloop
1710 .difline LDA lineno:CMP lineno2
1720 LDA lineno+1:SBC lineno2+1
1730 BCC lineless:LDA #ASC("M")
1740 JSR osasci:JSR osnewl
1750 LDA#1:STA pointer
1760 JSR listline2:JMP mainloop
1770 .lineless JSR listline
1780 LDA #ASC("F"):JSR osasci
1790 LDA#1:STA pointer
1800 JSR osnewl:JSR osnewl
1810 JSR incptr:BCC mloop3
1820 :
1830 .incptr LDA linelen
1840 CLC:ADC lineptr
1850 STA lineptr:BCC incptr2
1860 INC lineptr+1:CLC
1870 .incptr2 RTS
1880 :
1890 .close LDA #0:TAY
1900 JSR osfind:JSR setbasic
1910 LDA pointer:BNE ext2:JSR message
1920 .ext2 LDA #15:JSR osasci
1930 LDX stack:TXS:JMP exit
1940 :
1950 .getabyte STX savex:STY savey
1960 LDA &FF:BMI close:LDY handle
1970 JSR osbget:BCS close:LDX savex
1980 LDY savey:STA (lineptr2),Y
1990 INY:RTS
2000 :
2010 .setbasic LDA #&BB
2020 LDX #0:LDY #&FF
2030 JSR osbyte:TXA:TAY:LDA #&97
2040 LDX #&30:JMP osbyte
2050 :
2060 .listline LDA#1:STA pointer:LDA #A
SC("M")
2070 .list2 JSR osasci:LDA #32
2080 JSR osasci:JSR setbasic
2090 LDY #0:STY quoteflg:JSR plineno+4
2100 LDA #32:JSR osasci:LDY #4
2110 .printloop LDA (lineptr),Y
2120 CMP #ASC("""")
2130 BNE notquote:EOR quoteflg
2140 STA quoteflg:LDA #ASC("""")
2150 .notquote LDX quoteflg
2160 BNE inquotes:JSR chkgoto
2170 BCC notgoto:STY savey
2180 JSR plineno:LDY savey:BNE check
2190 .notgoto JSR ptoken
2200 INY:BNE check
2210 .inquotes JSR pchar:INY
2220 .check CPY linelen
2230 BNE printloop:JMP osnewl
2240 :
2250 .listline2 JSR swapdata
2260 LDA#1:STA pointer:LDA #ASC("F")
2270 JSR list2:JSR osnewl
2280 :
2290 .swapdata
2300 LDX lineptr:LDA lineptr2
2310 STA lineptr:STX lineptr2
2320 LDX lineptr+1:LDA lineptr2+1
2330 STA lineptr+1:STX lineptr2+1
2340 LDX lineno:LDA lineno2
2350 STA lineno:STX lineno2
2360 LDX lineno+1:LDA lineno2+1
2370 STA lineno+1:STX lineno2+1
2380 LDX linelen:LDA linelen2
2390 STA linelen:STX linelen2:RTS
2400 .cmd OPT FNequs("*COMPARE ")
2410 :
2420 .message
2430 LDY #255:.meslp INY:LDA text,Y
2440 JSR osasci:BNE meslp:JSR osnewl:RT
S
2450 .text OPT FNequs("Files match")
2460 OPT FNequb(0)
2470 ]:NEXT:ENDPROC
2480 :
2490 DEF FNequb(byte)
2500 ?P%=byte:P%=P%+1:=opt
2510 :
2520 DEF FNequs(A$)
2530 $P%=A$:P%=P%+LEN(A$):=opt
2540 :
2550 DEF PROCchecksum:S%=0
2560 FOR J%=start TO P%-1:S%=S%+?J%
2570 NEXT:ENDPROC
2580 :
2590 REM Basic-I entry points and check
sum value
2600 DATA &B571,&B53A,&98F1
2610 DATA &BE88,&97B6,&8A99,64495
2620 REM Basic-II entry points and chec
ksum value
2630 DATA &B558,&B50E,&991F
2640 DATA &BE6F,&97E7,&8AF6,64125
2650 REM Basic-IV entry points and chec
ksum value
2660 DATA &BD98,&BD37,&A081
2670 DATA &BDE5,&9B26,&8F86,64277
```

Thanks are due to Roger Cullis for his assistance with this program.

TELETEXT SUPER ROM

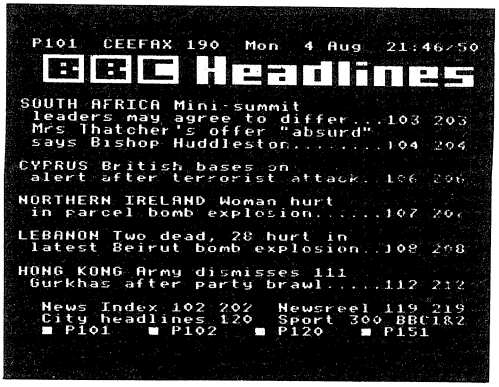
James Fletcher assesses the new BBC Teletext ROM, for use with Acorn's Teletext Adaptor.

Product: Advanced Teletext ROM
Producer: BBC Publications,
P.O. Box 234,
London SE1 3TH.
Tel. 01-407-6961 for credit card orders.
Price: £7.95 inc. VAT and p&p.

In the July issue of BEEBUG we discussed the many interesting things that you can do with a teletext/telesoftware adaptor for your BBC micro, and although I criticised several aspects of the way the adaptor from Acorn worked, I suggested that it was probably the best-buy for those who are not able or don't want to write their own software before making use of the system.

It came as something of surprise, then, that on the very day that the July BEEBUG dropped through my letterbox, BBC Publications announced that a new ROM was available to replace the existing one in the Acorn teletext adaptor. Since it costs only £7.95 and is available over the telephone if you give your credit card number, it didn't take long to acquire one, although BBC Publications did tell me that they hadn't yet got any in stock when I initially rang in response to a teletext page saying that the new ROM was 'now available'.

Called the Advanced Teletext System (ATS), the new 16K ROM is a direct replacement for the existing TFS ROM, and it really does seem to do all the things that the original system should have done! The ROM is accompanied by a user-guide and keystrip, and BBC Publications claim that it is compatible with all BBC micros, including the Master series. It should work with the ADFS as well as with other Acorn-compatible filing systems.



As with the existing unit, standard teletext pages from the four television channels can be displayed, but the new software now allows you to change channel at the touch of a key, and to select individual pages or sub-pages at will. There is an automatic facility for instantaneously calling up the previous page that you were watching. Also, for the first time, the unit allows you to make use of the linked-pages broadcast on CEEFAX, which makes selecting any desired page much easier.

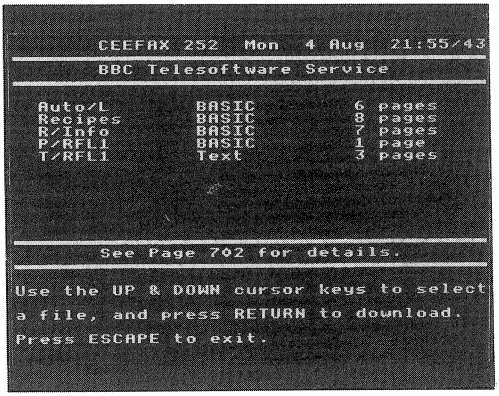
The most important changes however, come when downloading telesoftware. No longer is the teletext system regarded as a separate filing system, so users are spared the problems of continually switching between disc and teletext filing systems, with all the confusion and complexity that this can give rise to.

To see a complete list of all the telesoftware available from the BBC you merely press a single key, and a detailed menu-catalogue appears. By using the cursor keys to point an arrow at the appropriate part of the menu, the wanted program can be downloaded without having to worry about commands such as "LOAD" or *LOAD, and it will automatically be saved to disc without having to type in the various addresses for loading and execution, which can become tedious with the existing system.

Another snag with the current TFS system is that once having typed in your downloading instructions, the screen goes virtually blank and you are given no clues

about the state of the downloading process. This can be a bit worrying when some programs take up to twenty minutes to download, and if you accidently type in the name of the program wrongly the screen merely displays "SEARCHING" for ever! The Advanced Teletext System is much better in this regard, continuously showing the number of blocks of data that have been satisfactorily captured, and the number that remain to be downloaded, in an easily-understood graphical format.

No longer is it necessary to load a software patch before downloading, and perhaps even more important for some BBC micro users is the fact that the new ROM uses only 256 bytes of RAM, 256Ø bytes less than the current TFS system! When the adaptor is switched off or disconnected no



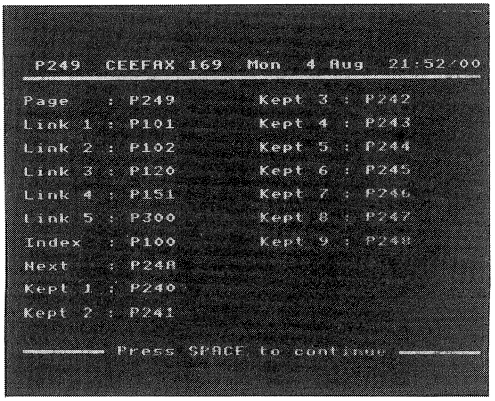
RAM space is claimed, which overcomes a problem that some users have had with the TFS ROM.

Since I make a great deal of use of the various 'interactive' teletext pages to build up databases, as described in last month's issue, I was worried that many of my existing programs would cease to work with the new ROM, but I needn't have worried; a TFS emulation mode has been included in the ATS ROM, so that existing software can carry on working as before. Some of the interactive telesoftware programs, broadcast previously, no longer correctly read the current date, but revised versions are being produced and re-transmitted.

As well as providing a far more user-friendly method of using telesoftware, the ATS ROM should enable the BBC to come up with a whole range of new telesoftware programs using the various new facilities. Already they have transmitted an interesting little program which allows a whole week's telesoftware programs to be downloaded off-air automatically, which really does make the whole process very much simpler than it was with the old system.

The 64-page instruction manual is spiral bound to match the usual 'user-guide' format, and contains a great deal of detailed but easily understood information, including sections on the teletext operating system commands and the relevant OSWORD command calls. It is a pity that there are only a very few examples of how to write your own programs to make use of the multiplicity of features that teletext can offer.

The whole package seems to be very good value for money; with the ATS fitted, the Acorn teletext adaptor can be highly recommended, and it is to be hoped that the new features will persuade many more Beeb users to take a look at telesoftware. In addition, BBC Publications are arranging for the new ATS ROM to be supplied as standard with new Acorn teletext adaptors when purchased from various retail outlets. Thus the ATS ROM looks set to become the standard that teletext has been seeking for so long.



by Mitch

ADVENTURE GAMES ADVENTURE GAMES

Title: The Quill
Supplier: Gilsoft, 2 Park Crescent, Barry, S. Glamorgan CF6 8HD
Price: £16.95 (Cas.), £22.95 (Disc)

It's taken three years, but at long last it's arrived in the BEEBUG dungeon. In the hands of Spectrum owners, The Quill has produced a long line of adventure games many of which have been notable successes. This product is the first truly professional adventure writer which I have seen for the BBC micro and Electron and it lives up to all my expectations.

The aim of The Quill is to enable users with limited programming ability to write a full size machine code adventure. This is achieved by menus, which enable the user to fill up tables of data which The Quill will act upon. The final game can then be saved and run independently.

Most adventures consist of a basic set of tables and 'flags' which control the movement of the player and the game's objects. It is relatively simple to automate the collection of this data from the user and formulate it into these tables. Where the difficulty arises for programmers and automatic writers alike is the creation of the logic code.

Somehow the game needs to 'know' that it must not allow the game player to 'Open the Green Door' unless firstly, 'The Dragon is dead', 'The player is carrying a copy of BEEBUG' and 'He is wearing a gold sequinned evening gown'. To this end The Quill provides a list of sixty words which are shorthand commands to cover most eventualities. (Including BEEBUG Adventurers who wear gold sequinned evening dresses!). Using these words, the user may construct a set of rules which he wishes the game to obey during the playing of the adventure. A bonus of this word list is that it can also serve to provide ideas previously unthought of by the user!

To ensure that you are not restricted in your game, The Quill also accepts any of the normal '*' commands in its list of logic statements (e.g. *LOAD). One further addition is that it accepts JSR commands, permitting you to jump to any machine code subroutine you may wish to include. By this means the only restriction to the complexity of your game is that of your own imagination and ability. By following the step-by-step instructions in the fifty page booklet you will be able to construct a small game in a few hours. A larger game which may normally have taken months to write will now take weeks.

And what of that bugbear of all adventures - graphics? The Quill advertises itself as text only, but in this respect it tends to undersell itself. True it has no facility to help draw screen shots but I suspect most owners already possess such a utility. To include these graphic shots within Quill adventures requires very little ability (I managed it in one hour!), so be wary of dismissing The Quill for this omission. Should you feel you must have this option built-in, I'm assured that a graphic addition is on its way for the autumn.

At any time during game creation you may run the game under diagnostics. This option has a panel displaying the values and states of various flags above the normal game display. This display enables the user to watch how the logic operates. Technical advice is also on hand for registered purchasers from Gilsoft.

All worthwhile software takes a time to master and this is no exception. However, this is a well thought out package and the long line of commercial games which have been produced from it for other machines is proof of its ability. I have little doubt that we will soon see a growing list of 'Quilled' adventures appearing for the BBC micro. Perhaps that adventure game you always wanted to write is now ready to be created?

Adverts for The Quill and the Graphic Adventure Creator from Incentive Software are currently appearing in magazines. Like many of you I am eagerly awaiting a copy of the G.A.C. to test its claims. Will The Quill be mightier than the coloured, fibre tip pen? Watch this space!

Printer Graphics (Part 2)

This month Alex Kang shows how to write a screen-to-printer dump, and describes a flexible dump program that you can tailor for your own printer.

The focus of this month's article is a multi-mode multi-tone screen dump with a windowing facility, and we will describe the use of this program before moving on to the general principles of writing screen dumps. This subsequent discussion, together with the fairly extensive in-program documentation should allow the dump to be modified for use with printers other than the Epson compatibles for which it was written.

Type in the accompanying listing, which contains a major machine code section, and save this to disc or tape. When you run the program, it will check to see if PAGE is set at &1300 or below. If not, it will ask you to reset it and reload the program. To do this, type:

PAGE=&1300 <Return>

before chaining the program.

The program makes two simple requests: it requires a mode number (0, 1, 2, 4 or 5), and a filename for the screen to be dumped. Once this is given, the file is loaded to the screen, and a small rectangular cursor will be seen flickering in the bottom left hand corner of the screen. This should be moved using the cursor keys (use Shift to speed up movement) to the required bottom left corner of the window to be dumped. Pressing the space bar will freeze the point, and then the process should be repeated to define the top right hand corner of the window. A final press of the space bar will freeze the point and commence the printout. Once the printout is complete, the printer will be reset, and the program will end. To dump another screen, simply re-run the program.

DIMENSIONING

There are two areas which require a certain amount of thought when designing printer dumps, and we will look at these in turn. The first is dimensioning. This is simply the problem of deciding how many dots should represent a given screen. Clearly the dot ratio is of great importance if we are to represent the computer screen faithfully. In practice it is rarely possible to obtain an exact match between the number of screen pixels, and the number of dots used to represent them on the printer. It is for this reason that circles get a little squashed in the printing.

The first step in the process is to select a printer graphics mode. We will use the Epson low speed double density mode, engaged with Esc 'L', which prints 960 dots per 8 inch line. We shall need to determine the number of horizontal dots per pixel for any graphics mode in which the dump is to operate, and this is calculated from ((number of dots per 8 inch line) DIV (horizontal resolution)). This gives the following:

Mode 0 : 1 dot per pixel
Modes 1,4 : 3 dots per pixel
Modes 2,5 : 6 dots per pixel

The screen dimensions, in graphics units, are 1280x1024, which gives a height to width ratio of 1:1.25. We require the printout to have a ratio as close to this value as possible. Each vertical dot occupies 1/72 of an inch, and the print width is 8 inches (960 dots per 8 inch line). So the print height should be 8/1.25=6.4 inches, which is equivalent to 72x6.4=460.8 dots high. However, in all graphics modes, the vertical screen resolution is 256 pixels. The closest we can get to 460.8 dots is 512 dots, making it 1 vertical pixel to 2 vertical dots. We thus end up with the following:

Mode	Width in dots/pixel	Height in dots/pixel
0	1	1
1,4	3	2
2,5	6	2

This data applies only to our chosen printer mode of 960 dots per 8 inch line. If your printer does not have this setting, choose the one closest to this and recalculate the values.

We must now decide upon the line spacing. The requirement for this is just

that there should be no overlapping, and no spaces between adjacent lines. Assuming we use 8 pin dot graphics, this is satisfied by setting the line spacing to 8/72 of an inch using Esc 'A' n where n=8.

COLOUR GRAPHICS

The other area which requires discussion is the way in which the different logical colours are to be represented on paper. We will look at the approach adopted in the program which accompanies this article.

Graphics modes 1, 2 and 5 use either 4 or 8 colours (8 because we do not include flashing colours - unless we've got flashing ink!). Since we have a 3x2 dot matrix per pixel for mode 1 and a 6x2 dot matrix per pixel for modes 2 and 5, we can design our own bit patterns for these pixel units to represent the various colours. In the program Flexidump, 8 sets of bit patterns have been designed on a 6x2 matrix for all the colour modes (see figure 1). So long as the first 4 sets have a significantly different pattern in the left 3 columns, these can be used for mode 1 too. Once you have designed the bit patterns, just calculate the values of each column as you would the two most significant bits of a byte, as shown in figure 1. You may substitute your own bit patterns for the ones used in the program.

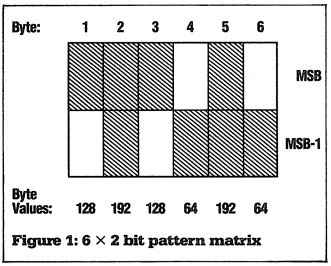


Figure 1: 6 × 2 bit pattern matrix

PROGRAM NOTES

We include here some specific notes on Flexidump. We have seen that in modes 1, 2, 4 and 5, each pixel is represented by a block 2 dots high. The print head is 8 dots high, so that we can print 4 vertical pixels simultaneously. Given the starting X and Y co-ordinates, we may print a line by reading the first column of 4 pixels, then printing it; incrementing the X co-ordinate to read the next column of 4 pixels, printing it, and so on. Each printed line means we have to decrement the Y co-ordinate by 4 pixels (i.e. 16 graphics units). The variable Xstep% holds the number of horizontal graphics units per pixel (i.e. 1280 divided by the horizontal resolution) and npb% represents the number of horizontal dots per pixel (equal to the number of printing bytes per pixel). D% is the number of horizontal dots within the printing window and thus represents the number of data bytes per line we have to send to the printer.

The assembler routine reads each column of four pixels from the screen, and prints out their corresponding bit patterns. The logical colour of each pixel is read with an OSWORD call with A=9 and its X and Y co-ordinates in a parameter block starting at &70. As each pixel in the column is read, the corresponding bit pattern is transferred to a pattern buffer. This is then right-shifted a certain number of times, and added to the bytes to be printed. This is illustrated in figure 2 for a mode 1 (3x2 bit pattern) dump. Once all four pixels have been read, shifted and added to the print bytes, the resultant print bytes are printed out. This process continues line by line from left to right and from top to bottom of the window to be printed.

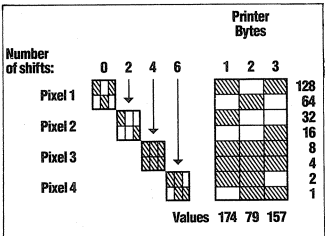


Figure 2: Read pixel bit patterns, shift and add to printer bytes.

With mode 0 however, things are slightly different. Because each pixel is represented by a single dot, we can read 8 pixels at once and print them directly. And since there are only two possible logical colours, we needn't bother about bit patterns. Instead we have a byte indicating the pixel value (zshift) with an initial value of 128 for the top pixel in the column. This is right-shifted for each pixel read in the column, and if the pixel is on, it is added to the data byte to be printed.

Next month we conclude this brief series with the BEEBUG Page Designer, an integrated package allowing the creation of full pages of mixed graphics and text.

```
10 REM Program Flexidump
20 REM Version B 0.3D
30 REM Author  Alex Kang
40 REM Beebug  October 1986
```

```
  50 REM Program subject to copyright
  60 :
 100 REM MODE VARIABLES:
 110 REM  Mode  Xstep%   npb%
 120 REM    0     2       1
 130 REM   1,4    4       3
 140 REM   2,5    8       6
 150 REM npb% is no.of dots/pixel width
 160 MODE 7:PRINTTAB(4,5)"FLEXDUM"'TAB(
4)"by Alex Kang 1986"
 170 IF PAGE>&1300 THEN PRINT'TAB(4)"Pl
ease reload program with PAGE":PRINTTAB(
4)"set to &1300":END
 180 REPEAT:INPUTTAB(5,10)"Screen mode:
 "md%:UNTIL (md%<3) OR (md%>3 AND md%<6)
 190 MODEmd%:?&75=md%:DIM name 25
 200 Ystep%=-16*(md%<>0)-32*(md%=0)
 210 Xstep%=-2*(md%=0)-4*(md%=1 OR md%=
4)-8*(md%=2 OR md%=5)
 220 npb%=-1*(md%=0)-3*(md%=1 OR md%=4)
-6*(md%=2 OR md%=5)
 230 ?&8F=Xstep%:?&8E=npb%
 240 PROCass:PROCbitpatterns
 250 PROCldp:PROCcrop
 260 REM No.of data bytes to be printed
 270 D%=?&8E*((U%-P%+1)/Xstep%)
 280 N1%=D%MOD256:N2%=D%DIV256
 290 REM Set line spacing to 8/72 in.
 300 VDU2,1,27,1,65,1,8:Y%=V%:REPEAT
 310 REM Line feed & set graphics mode
 320 VDU1,13,1,27,1,76,1,N1%,1,N2%
 330 CALL dump
 340 REM update y start coord
 350 Y%=Y%-Ystep%:?&78=Y%MOD256:?&79=Y%
DIV256
 360 REM reset x coord for new line
 370 ?&70=P%MOD256:?&71=P%DIV256
 380 UNTILY%<=Q%
 390 VDU1,27,1,50,3:VDU23,1,1;0;0;0;
 400 END
 410 :
1000 DEFPROCbitpatterns
1010 FORI%=0TO47:READP%
1020 I%?bitpatn=P%
1030 NEXT:ENDPROC
1040 REM Bit pattern data in order of
1050 REM logical colours 0-7. May be
1060 REM user modified.
1070 DATA 0,0,0,0,0,0
1080 DATA 192,192,192,192,192,192
1090 DATA 128,0,64,64,0,128
1100 DATA 0,0,64,64,0,0
1110 DATA 128,0,64,0,128,0
1120 DATA 128,64,128,64,128,64
1130 DATA 192,64,192,192,64,192
1140 DATA 192,0,128,64,0,192
1150 :
1160 DEFPROCass
1170 REM Code assembled at &900
1180 dump=&900
1190 xcoord=&70:ycoord=&72:pixcol=&74:m
ode=&75:ndata=&76:ystart=&78:xmax=&7A:ym
in=&7C:bitpnter=&7E:patbuf=&80:printbyt=
&86:zshift=&8C:pixelnum=&8D:npbyte=&8E:x
step=&8F:osword=&FFF1:oswrch=&FFEE
1200 FORpass=0TO2 STEP2
1210 P%=dump
1220 [OPTpass
1230 \ transfer y start coord.
1240 \ to y-coord counter.
1250 .initdump LDA ystart:STA ycoord
1260 LDA ystart+1:STA ycoord+1
1270 LDX#5:LDA#0
1280 \ set printer data bytes and
1290 \ bit pattern buffers to zero.
1300 .settozero
1310 STA printbyt,X:STA patbuf,X
1320 DEX:BPL settozero
1330 \ Get screen mode, branch if 0
1340 LDA mode:BEQ mode0
1350 \ Routine for modes 1,2,4,5
1360 .notmode0 LDA#0:STA pixelnum
1370 .rep JSR readpixel
1380 \ get bit pattern of pixel colour
1390 \ and transfer to pattern buffer.
1400 .getpat ASL A:CLC:ADC pixcol
1410 ASL A:STA pixcol
1420 LDA #bitpatn MOD 256
1430 CLC:ADC pixcol:STA bitpnter
1440 LDA #bitpatn DIV 256
1450 ADC#0:STA bitpnter+1:LDY#5
1460 .tobuf
1470 LDA (bitpnter),Y:STA patbuf,Y
1480 DEY:BPL tobuf
1490 \ Shift pattern buffer bytes?
1500 LDY pixelnum:BEQ trfbp
1510 .shft LDX#5
1520 .sh1 LSR patbuf,X:LSR patbuf,X
1530 \ Right shift all bytes twice
1540 DEX:BPL sh1
1550 \ Repeat shift for Y times
1560 DEY:BNE shft
1570 \ Transfer pattern buffer to
1580 \ printer data buffer.
1590 .trfbp LDX#5
1600 .trf1 LDA printbyt,X
1610 CLC:ADC patbuf,X:STA printbyt,X
1620 DEX:BPL trf1
1630 INC pixelnum
1640 \ all 4 pixels in column read?
1650 LDA pixelnum:CMP#4:BEQ p1
1660 JSR decy
1670 \ check for y minimum range
1680 LDA ycoord:CMP ymin:BEQ y1:BCS rep
1690 .y1 LDA ycoord+1:CMP ymin+1:BEQ p1
:BCS rep
1700 \ print and increment x coord
1710 .p1 JSR print:JMP incx
1720 :
1730 \ routine for mode 0
```

```
1740 .mode0 LDA#128:STA zshift
1750 LDA#8:STA pixelnum
1760 .repl
1770 \ is logical colour 0?
1780 JSR readpixel:BEQ deccn
1790 \ if not add pixel to data byte
1800 LDA zshift
1810 CLC:ADC printbyt:STA printbyt
1820 \ shift pixel value and
1830 \ update pixel number counter.
1840 .deccn
1850 LSR zshift:DEC pixelnum:BEQ p1
1860 JSR decy
1870 \ check for y minimum range
1880 LDA ycoord:CMP ymin:BEQ y2:BCS rep
1
1890 .y2 LDA ycoord+1:CMP ymin+1:BEQ p1
:BCS repl
1900 \ print it
1910 JMP p1
1920 \ increment x coord. by xstep and
1930 \ check for x maximum range.
1940 .incx LDA xcoord
1950 CLC:ADC xstep:STA xcoord
1960 LDA xcoord+1:ADC#0:STA xcoord+1
1970 LDA xcoord:CMP xmax:BCC notmax
1980 LDA xcoord+1:CMP xmax+1:BCC notmax
1990 RTS
2000 .notmax JMP initdump
2010 \ read logical colour of pixel.
2020 .readpixel LDX#xcoord:LDY#0
2030 LDA#9:JSR osword
2040 LDA pixcol:RTS
2050 \ decrement y coord. counter by 4
2060 .decy
2070 LDA ycoord:SEC:SBC#4:STA ycoord
2080 LDA ycoord+1:SBC#0:STA ycoord+1
2090 RTS
2100 \ executes VDU1,data
2110 .vdu LDA#1:JSR oswrch
2120 TXA:JSR oswrch:RTS
2130 :
2140 \ outputs data bytes to printer
2150 .print LDY#0
2160 .prl LDX printbyt,Y:JSR vdu
2170 INY:CPY npbyte:BNE prl:RTS
2180 :
2190 \ bit patterns stored here
2200 .bitpatn
2210 ]:P%=P%+48:[OPT pass
2220 ]
2230 NEXT:ENDPROC
2240 :
2250 DEFPROCbox(K%,L%,M%,N%,O%)
2260 *FX19
2270 MOVEK%,L%:PLOTO%,M%,L%:PLOTO%,M%,N
%:PLOTO%,K%,N%:PLOTO%,K%,L%:ENDPROC
2280 DEFPROCmove(K%,L%,M%,N%)
2290 LOCAL s%,s1%
2300 s%=Xstep%+(-32*INKEY-1):s1%=4+(-32
*INKEY-1)
2310 X%=X%+s%*(INKEY-26-INKEY-122)
2320 Y%=Y%+s1%*(INKEY-42-INKEY-58)
2330 sp%=INKEY-99
2340 IF X%<K% X%=K%
2350 IF X%>L% X%=L%
2360 IF Y%<M% Y%=M%
2370 IF Y%>N% Y%=N%
2380 REPEATUNTIL NOT INKEY-99
2390 *FX15,1
2400 ENDPROC
2410 DEFPROCcrop
2420 X%=0:Y%=0:REPEAT
2430 PROCmove(0,1248,0,992)
2440 PROCbox(X%,Y%,X%+31,Y%+31,6):PROCb
ox(X%,Y%,X%+31,Y%+31,6)
2450 UNTILsp%:VDU7:*FX15,1
2460 P%=X%:X%=X%+31:Q%=Y%:Y%=Y%+32
2470 REPEAT
2480 PROCmove(P%+31,P%+1279,Q%+31,Q%+10
23)
2490 PROCbox(P%,Q%,X%,Y%,6):PROCbox(P%,
Q%,X%,Y%,6)
2500 UNTILsp%:VDU7
2510 IF Y%>1023 Y%=1023
2520 IF X%>1279 X%=1279
2530 U%=X%:V%=Y%:U%=U%+1:Q%=Q%-4:IF Q%=
-4 THEN Q%=0
2540 REM Store x start coord.
2550 ?&70=P%MOD256:?&71=P%DIV256
2560 REM Store y start coord.
2570 ?&78=V%MOD256:?&79=V%DIV256
2580 REM Store y minimum range coord.
2590 ?&7C=Q%MOD256:?&7D=Q%DIV256
2600 REM Store x maximum range coord.
2610 ?&7A=U%MOD256:?&7B=U%DIV256
2620 ENDPROC
2630 DEFPROCoscli:LOCAL X%,Y%
2640 X%=name MOD256:Y%=name DIV256
2650 CALL &FFF7:ENDPROC
2660 DEFPROCldp
2670 PRINTTAB(0,5)"LOAD PICTURE:"''"Fil
ename: ";:INPUT p$:C%=OPENUPp$:IF C%=0 V
DU7:PRINT'"NO SUCH PICTURE!"'"PRESS <SPA
CE>..":IF GET CLS:GOTO2670
2680 IF md%>=0 AND md%<=2 md$=" 3000" E
LSE md$=" 5800"
2690 CLOSE#C%:$name="*LO. "+p$+md$:PROC
oscli:VDU23,1,0;0;0;0;:ENDPROC
```

This program will run on a Compact provided that Basic IV is loaded from the Welcome disc and used instead of the built-in ROM Basic.

If you purchased a Master 128 or ET between January-May 1986 please read on.

This is an important message for users of BBC Master Series micros purchased between January and May 1986. You should read this announcement carefully then contact your Acorn dealer if you have not already done so.

It has been found that when the battery in the above models is close to exhaustion, the microcomputer may attempt to recharge it. This is contrary to the recommendations of the battery manufacturer.

An upgrade kit which prevents this happening has been developed and is now available from any Acorn dealer at no cost. Fitting can be carried out easily either by you or free of charge by your supplier. This will not invalidate your guarantee.

If you have difficulty in obtaining a kit, please call 0223-214411 and ask for Department A2.

In the meantime, we suggest you follow standard electrical appliance guidelines and keep the power switched off when your microcomputer is not in use.

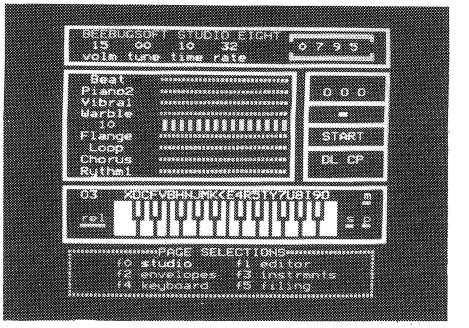
We apologise for any inconvenience that this upgrade will cause but we hope that Acorn's customers will be assured by our desire to maintain a high standard of product in the field.

Acorn
The choice of experience.

STUDIO 8

This pack contains more than 20K of machine code and converts your BBC micro into a four voice synthesiser with an eight track digital recorder and a rhythm and drum machine.

- Select any one of the 32 instruments by name. Play them from the keyboard with sliding fader controls for level mixing
- Set keyboard to auto-sustain, mono or polyphonic, with optional split-keyboard operation
- Play the keyboard; alter the tonal effects; see the amplitude pulsate on the sliding faders
- Build up a drum backing on the drum machine
- Build up a rhythm accompaniment on the rhythm machine
- Synchronise the two
- Set the 8 track recorder to record, and watch the digital tape counter
- Use the studio mixer to add an accompaniment from a different instrument played in real-time on the synthesiser



- Press "Rewind"; watch the tape indicator
- Press "Stop" at any point: then press "Play" to hear the mix

This is a truly exciting real-time studio system which will give hours of entertainment. It does not require musical proficiency and will addict you with its ease of use and amazing results.

Simple instructions are given to add an external speaker and keyboard.

Examples of the music that can be created with **STUDIO 8** are included on the disc/tape.

OTHER FEA

- **Envelope definer** — 16 en… …d using a full screen editor; and may be slow…
- **Instrument d**… …2 instruments with up to 16 envelopes acces…
- **Music editor** — A …creen editor to edit music recorded on the system. Includes printout facility

Now compatible with the Master 128 & Master Compact!

Disc	**£22.00**	Members	**£16.50**
Tape	**£17.00**	Members	**£12.75**

New High Scores

Supplier	Game	Score	Player
ACORNSOFT	Carousel	52070	E.Somerville
ACORNSOFT	Rocket Raid	206270	K.Butler
ALLIGATA	Tarzan	275200	K.Butler
ATARISOFT	Donkey Kong Jnr.	92700	K.Butler
BEEBUG MAG	Manhole	260	Miss Ashford
BUG-BYTE	Oblivion	7235	K.Butler
ELITE	Commando	104400	K.Butler
ICON	Bug Eyes	62380	K.Butler
IMAGINE	Yie-Ar Kung Fu	804700	M.Reardon
KANSAS	Moon Buggy	48840	K.Butler
MICRO POWER	Castle Quest	2260	K.Butler
MICRO POWER	Danger! UXB	1271860	E.Somerville
SOFTWARE INV	Alpha Centauri	45400	K.Butler
SOFTWARE INV	Blitzkrieg	160000	E.Somerville
SOFTWARE PROJ.	Manic Miner	27342	K.Butler
SUPERIOR SOFT	Karate Combat	38763	K.Butler
SUPERIOR SOFT	Repton 2	18200	R.Watts
ULTIMATE	Atic Atac	90%	K.Butler
US GOLD	Beach Head	122200	E.Somerville
US GOLD	Tapper	314425	K.Butler
WATFORD ELEC.	Crawler	70994	K.Butler

BEEBUG WORDPROCESSING OFFERS

INTERWORD

The new wordprocessing Rom from Computer Concepts. Fully integrates with the other programs in their Rom Link suite.

ROM-LINK

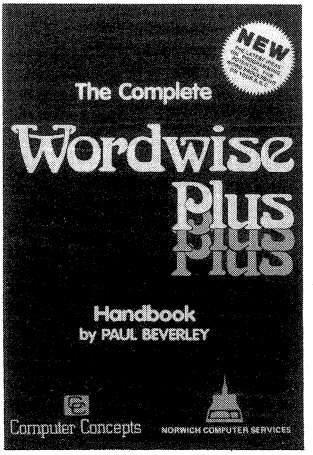
The ROM-LINK suite, of which INTER-WORD is an important part, is a uniquely flexible set of integrated ROMs, compatible with all models of the BBC micro.

INTER WORD

Normal Price £56.35
Beebug Price £47.00 plus £1.00 p&p
Wordwise Plus owners may quote their registration number to claim a further £3.00 discount.

The Complete Wordwise Plus Handbook
By Paul Beverley, Norwich Computer Services.

An excellent handbook for anyone using Wordwise Plus. Over 400 pages of information, advice, hints, tips etc etc.

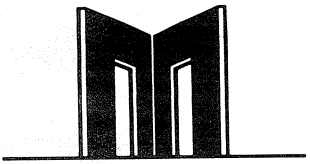


As reviewed in Beebug Aug/Sept. 1986

We are able to offer the book and accompanying disc for the price of £19.00 plus £1.50 p&p. Normal price £25.00 inc p&p.

Available from Beebug Retail, Dolphin Place, Holywell Hill, St. Albans, Herts. AL1 1EX.

TRADE IN YOUR BEEB FOR A MASTER 128 OR MASTER COMPACT



For a trial period we are offering Beebug members the opportunity to trade in their BBC Micros towards the purchase of a new Master.

Depending on the age of your computer, we will allow a trade-in discount of up to £225 off the price of a new BBC Master computer. This would bring the purchase price down from £489 for a Master 128 to £264.

The discounts are as follows:

BBC Micro issue 4 or older without DFS **£125**

BBC Micro issue 4 or older with DFS **£175**

BBC Micro issue 7 or newer without DFS **£175**

BBC Micro issue 7 or newer with DFS **£225**

By DFS we mean an Acorn or Watford single density DFS. Computers must obviously be in good working condition to qualify for this offer.

If you wish to take advantage of this service please write to us or telephone the number below. Do not just send your machine, as we are only able to accept orders subject to availability.

Second Hand BBC Micros For Sale

As a result of this service we will also be able to offer a number of second hand BBC Micros for sale.

Machines will be thoroughly serviced and offered for sale with 3 months guarantee, prices will start from £180. Please write or phone to discuss your requirements.

Beebug Retail, Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX
Telephone: 0727 40303

Discounts

BEEBUG has arranged discounts for members at a number of retail outlets who supply computer books, software, hardware and services. We are continually negotiating further discounts.

It is advisable to telephone before placing an order, to check availability. Members should simply quote their membership number with their order, though members taking discount will not necessarily be given credit card facilities, (you must check this). We are not acting for these companies, nor receiving payment from them, and cannot be held responsible for their services.

AKHTER computer group,
28/29 Burnt Mill, Harlow,
Essex CM20 2HJ.
Tel: 0279 443521

5% discount to members for cash or cheque with order.

ALLM Books,
21 Beechcroft Road,
Bushey, Herts.
Tel: 0923 30150

10% discount for members of BEEBUG and ELBUG the Small Computer Program Index and other database and library software.

Appleby, Myers and Clarke,
60 Market Street,
Watford, Herts.
Tel: 0923 31960

Free postage on books.

AVC Software,
PO Box 415, Harborne,
Birmingham B17 9TT

25% off 'Learning with the Beeb', and 'Learning with the Elk' discount price: £1.50 for BEEBUG and ELBUG.

Calancraft Ltd.,
134a St. John's Road,
Woking, Surrey GU21 1PS
Tel: 0932 65284

From 5% to 15% on microcomputer equipment, workstations, disc drive cases & PSU's and software.

Computer Holidays,
37 University Road,
Highfield, Southampton
Tel: 0703 558621

£5 discount for BEEBUG and ELBUG.

Danum Computer Systems,
17 East Laith Gate,
Doncaster,
S. Yorkshire DN1 1JG
and Computer Room,
Unit 12, Market Hall,
Barnsley, S. Yorks.

Discount prices on request.

East Riding Computers,
24 Sherbuttgate Road
South,
Pocklington,
York YO4 2YB
Tel: 07592 4387

Discount prices to members on request.

Edsoft,
12 Cotmore Close,
Brixham TQ5 0EF

10% discount on Software

K. Edward,
54 Greyham Court,
Dalclaverhouse,
Dundee DD4 9DQ

5% discount VDU Support Stand kits.

---

fFolders,
10 Weston Lane,
Bridgetown,
Totnes, Devon.

15% discount on function key labels.

FUREY Enterprise Products,
7 Manor Park,
Froxfield, Wilts. SN8 3LF
Tel: 0488 83006

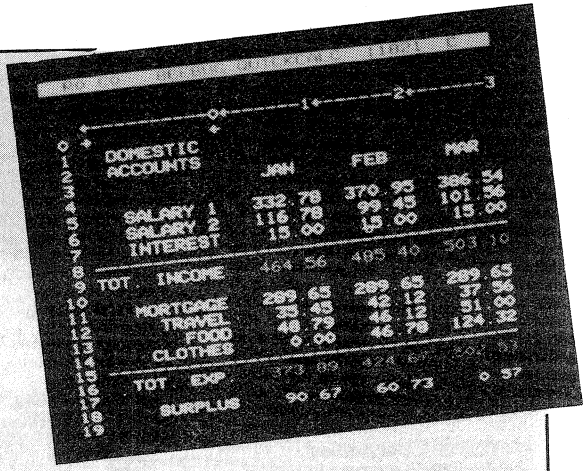
10% Discount.
BBkey Function key booklet – £2
BBtype word processor – £23
View/Wordwise Reformatter – £2

→ 34

Quickcalc

AN EASY-TO-USE SPREADSHEET

QUICKCALC is an easy-to-use, extremely powerful spreadsheet supplied on cassette or disc. It will bring sophistication and computing power to a wide range of different number-based activities.



Use it for your personal accounts, school marks, stock control, sales, VAT, and general financial planning. It allows rapid and automatic calculation of all kinds, and permits forward planning "what if" analyses to be effortlessly performed.

QUICKCALC offers load, save, print, goto, re... and many other functions and is supplied with ... manual and function key strip. 6502 second proces...

Now Compatible with the Master 128 & Master Compact!

Dumpmaster

PRINTER DUMPS

A clever piece of software which produ... machine code, tailor-made screen dumps ... any of the different types of printers listed below. DUMPMASTER uses up to 8 shades and will accurately copy your screen in any of the graphics modes, and even the teletext mode.

Once created your tailor-made dump can be called as required or appended to your own Basic program.
There is even a special 'Snapshot' facility which will produce screen dumps from a game or other program, at the press of a key.
(As long as the program does not use the same work area.)

DUMPMASTER II

Now in Rom supporting the following printers

- ANADEX DP 9000 series
- BROTHER M1009
- * BROTHER HR5
- CANON A-1210
- * CANON PJ1080A
- CANON PW series
- COSMOS 80
- DATAC 109V
- ENSIGN 1650
- EPSON all
- EPSON FX.LX
- FACIT 4510
- GEMINI
- * IDS-480
- INTEGREX 132 COLOUR
- KAGA TAXAN
- MANNESMANN TALLY MT80
- MANNESMANN TALLY MT160
- MICRO-P MP-165
- NEC PC 8023
- * OLIVETTI JP101
- PANASONIC KX-P series
- * QUEN DATA DP100
- SEIKOSHA GP80
- SEIKOSHA GP100
- SEIKOSHA GP250
- * SEIKOSHA GP550
- SEIKOSHA GP700A
- SHINWA CTI CP80
- STAR DP-8480
- STAR all others
- * TANDY CGP115
- * TANDY DMP-100

* Only available on Rom version.

←32

Globe Book Services,
Houndmills,
Basingstoke,
Hants. RG21 2XS

10% Discount on 'Assembly Language Programming for the BBC Micro' by I. Birnbaum.

Hermes Computer Products,
23 Durnford Close,
Norden, Rochdale OL12 7RX

10% discount + free p.&p. on a graphics design pad.

ITM Ltd.,
E3 New Enterprises,
S.W. Brunswick Dock,
Liverpool L3 4AR

Rotaview Professional videotext system. Usually £70 + VAT + pp. Now with £5 discount.

Kingsley TV Services,
40 Shields Road,
Newcastle Upon Tyne
Tel: 0632 650653

£10 discount off their range of Monitors and TV/ Monitors.

Microage,
135 Hale Lane,
Edgware, Middx. HA8 9QP

5% discount off some disc drives.

Microguides,
St. James House,
105-113 Broadway,
London W13 9BL

10% discount for 3 or more of their keyboard overlays.

Microtest Ltd.,
18 Normandy Way, Bodmin,
Cornwall PL31 1EX
Tel: 0208 3171

10% discount on Software, and 5% discount on Hardware.

Microware,
637 Holloway Road,
London N19

10% discount on DDFS. Other discount prices as shown in Microware advert.

Micro Computer World,
1 Lane Close, London
NW2 6QZ
Tel: 01-452 0893
Write for free list.

Discount on Software for BBC & Electron. Discount on Hardware Joysticks and Accessories.

Micro Supplies.
Oxford House, Faldouet,
St. Martin, Jersey, C.I.
Tel: 0534 55743

5% discount on diskettes.

Mine of Information,
(Mail Order),
1 Francis Avenue,
St. Albans,
Herts. AL3 6BL
Tel: 0727 52801

Computer Books 5% discount.

Off Records,
Computer House,
58 Battersea Rise,
Clapham Junction,
London SW11 1HH

25% off diskette purchases.

Opus Supplies,
55 Ormside Way,
Holmethorpe Industrial Estate,
Redhill,
Surrey RH1 2LW

Generous discounts on all Opus Products for BEEBUG members

Oxford Computer Supplies,
PO Box 133,
Oxford OX4 3NB
Tel: 0865 716137

7.5% discount on discs, printer paper & word-processing paper.

P.M. Communications,
47 Wickentree Lane,
Failsworth,
Manchester M35 9AY
Tel: 061 688 7444

Large discounts on disc drives, diskettes, and power supplies.

Shinecrest,
Unit 14, Dicker Mill,
Hertford, Herts. SG13 7AE
Tel: 0992 51977

10% off their home computer work station to ELBUG and BEEBUG members.

→38

Personal Ads

As from the October issue of BEEBUG, we are offering all BEEBUG members the opportunity to place personal ads free of charge in the magazine (see Editorial Jottings). Please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any ad if necessary. We shall still continue to accept members' business ads at the rate of 30p per word (inclusive of VAT) and these will be featured separately in future. Please send all ads (personal and business) to MEMBERS' ADS, BEEBUG, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX. The normal copy date for receipt of all ads will be the 12th of each month.

BBC B, DFS, ADFS, Solidisk 256K 4 meg board £400. ROMs: Exmon, Graphics £10 each. WW+ £25. Light pen £10. Various related books. Tel: Langport 250130.

Cumana CD200 dual 40 track, single sided, slimline disc drive with P.S.U £125. Tel: Potters Bar (0707) 54311 after 7pm.

Cassette games software for BBC Model B for sale, all original and popular titles, worth £175 - offers on (021) 474-4156.

Wordwise Plus + all manuals £39. Telephone Epping 76672.

Disc drives, dual, double sided 40 track 400k, Beeb powered, with cables, manuals. Owner upgrading, £99. Viewsheet user guide, £6. Modem, acoustic with ROM for Prestel, £24. Aries, Wheathampstead 3937.

Hobbit floppy tape drive, with manual, interfaces, zero-option device and 25 minicassettes. Offers to B.C.Sexton. Tel: (0742) 582931.

Music 500, issues 1 and 2 tapes + 1 year Ample User subscription £100 ono. Ikon Ultradrive 1.04 UFS + manual and 18 mini cassettes £80. M.Lee (0535) 603277.

I have 14 adventure cassettes, all originals! only £2 each! Also Fileaid ROM and disc, AID assembler ROM, 4 slot ROM board and Starstick II ROM (complete with joystick) at £10 each. Contact: D.Shepherdson, 3 Tarn Villas, Cowpasture Road, Ilkley, West Yorks. LS29 8RH, or tel: (0943) 609866.

Beeb, drive, many accessories, offers? Ring for details: R.Farmer (0226) 200719.

Little used Cumana CS100 100K 40T disc drive in original packaging £100. G.Legg (0273) 832548 evenings/weekends.

Wordwise Plus, unused. ROM, manuals and registration card. Obtained as part of the Solidisk 256k upgrade. £25. Marlow 73737.

Philips green monitor 80, boxed, 1 year old £49. View wordprocessor £29. Viewsheet spreadsheet £29. UDM double density DFS Acorn compatible £19. Dot print plus NLQ typeface enhancement ROM for Epson printers £15. All above with keystrips, manuals etc. G.Brown 01-341 2187.

Wordwise Plus wordprocessor chip, with less than a million words on the clock, plus manual and introduction. £25 ono. Tel: 01-328 4897.

Z80 second processor unused. All software, backup copies and manuals - perfect order. £300.00. Call Mr. Sykes on 01-670 8523 or 01-240 2040 or 01-837 6273.

Acorn 500 music system. £75 ono. Tel: 01-995 9250.

BBC B. Purchased March 85 with DNFS and RAM/ROM expansion board. Boxed £285. Ring St.Albans 66327/53695.

For Sale. Spellcheck II by Beebugsoft, still in original package, complete with manual, fitting instructions and 40 track dictionary disc. £20 ono. Tel: (0695) 74091.

→36

Magazines for sale: Simon Howard, 9 Springmeadow Lane, Uppermill, Oldham, Lancs. OL3 6EP. S.A.E for a list.

Wordwise Plus £22, Intersheet £22, Printmaster (Epson) £13. Acorn second processors, 6502 £95, Z80 £180. All original, as new, with software and manuals. Phone L.Evans (0454) 778503 evenings.

Beebugsoft Sleuth ROM as new: £20. Write P.Jacobs, 15 Cheriton Road, Winchester, Hampshire SO22 5EQ.

For Sale: Aries B20/B12 with 16K RAM £65. ROMs - Disc Doctor £20, Caretaker £20, Communicator £25, Graphics Extension £20. Also a number of books at half price or less. Tel: (0763) 42593 evenings.

BBC B + DFS £170; Aries B32 and B12 £100; Centronics 737 printer £60; Solidisk 32K + W1770 and DFS £70; Watford ROM board £10; Prism modem 1000 £20; BROM £15; Sleuth £15; Wordease £10; Disc Doctor £10; Beebcalc £5; Beebugsoft Toolkit £5; Discmaster + The Key £5. All OK and with manuals. P&p extra. Phone David Pears, Upminster (04022) 29912.

For Sale. Mini Office by Micro User. 40 track disc, still in original package complete with instructions. £4.50. Tel: (0695) 74091.

Software for sale. Original games, adventures, music or business programs, complete with documentation where applicable. S.A.E. for full list and prices. R.J.Follett, 26 Arbor Lane, Winnersh, Berks. RG11 5JD.

Acorn Teletext receiver, with ROM and manual, mint condition £80. Watford sideways ROM board (MkII) with 16K sideways RAM fitted £35. Opus double density disc controller board, complete with latest ROM and manuals £60. All in 100% working order, for sale due to upgrading. Tel: (02814) 5332.

Beebcalc ROM (Acornsoft) + manual and cassette, £15 ono. Tel: L.Errington (031) 556-9093.

Ready to buy back copies of A&B Computing for the BBC, from issue 1 up to and including the October 1984 issue. Copies must be in good condition. Separate copies considered. Please state cost and send to Revd. J.Pace, 91 Church Street, Msida, Malta G.C.

Torch Z80 disc pack with Perfect software, hardly used £395, or nearest offer. Tel: (0792) 204166 anytime.

Acornsoft ROMs for sale:- View 3 £30, Viewsheet £20, Basic II £10. Phone (04747) 7724.

BBC bits for sale. Sir ROM board with 16K RAM £25, DNFS £10, Basic II £10, Cheetah Sweet Talker £15, MOS 1.2 £5. Tel: (02572) 78286.

BBC B, O.S 1.2, Basic II, Acorn DFS, Exmon, 40 column thermal printer, joysticks, lightpen, 2 cassette recorders, software on tape (including Forth and Lisp), magazines and books. £350. Tel: Leeds 781911 evenings.

For sale: ATPL sideways ROM board for BBC B, Acornsoft View 2.1 and Viewsheet. £100 the lot. Tel: 01-882 1337.

Acorn Z80 processor with discs and manuals £250. Learmonth, 99/1 Canongate, Edinburgh. Tel: 031-556 1794.

ROMs: Toolkit plus with manual and registration card £29. Acorn DFS £10. Sold separately or £35 both or will swap for other Master compatible ROMs. Tel: (0202) 761936.

Wordwise Plus ROM £30, Opus 1770 DDOS £30, Replay ROM £20, Beebed ROM £12, Joystick ROM and Quickshot II £13, ADT ROM £18. Tim Riddle, 61 Midhurst Road, Fernhurst, Haslemere, Surrey. GU27 3EN.

Serial printer, tractor feed, 30cps, 118 columns, with paper and BBC cable £30. Tel: 061-439 9665.

Classified Ads

Problems? try R-SOFT utilities!
1. HOW-TO: An essential collection of software and instructions for frustrated new disc owners who want to move their programs to disc.
3. ROMPULL + TAPEDUMP.
4. AUTOMATIC DISC MENU: Includes sideways-RAM version, can boot from ROM, works with 2nd 6502.
6. SWROM*: Puts your Basic / machine code programs in ROM format.
7. RFS-GENERATOR: Generates ROMs for the *ROM filing system. This does not use the DFS workspace and is an ideal tool to run nasty programs from disc.
All above packages £5.00 each.
13. D-MASTER-V2: Superb disc backup program; needs 8271 and swram; £8.00; on 16K EPROM £12.00. Upgrade for D-MASTER owners: £500/£9.00.
16. OPUS SD/DD/Challenger: Disc menu, disc editor, disc indexer, etc. Please enquire for further details.
17. D-EDITOR: based on D-MASTER-V2: will edit any disc D-MASTER-V2 can backup! £8.00/£12.00.
Please send SAE for full list.
R-SOFT, 22 Marriotts Close, Felmersham, Bedford MK43 7HD. Tel: (0234) 781730.

CONVERT your 80T drive to 40/80 switchable. MkII unit (cased) simply plugs in - £19. MkI kit as above but requires soldering iron, fits inside drive - £11. Sideways RAM system 16K with software to load/save ROMs - £25. All prices include VAT and p&p. Computer repairs, competitive rates. Please telephone for further information. Dept B1, MAGUS Electronics, 33 Pool Lane, Winterley, Sandbach, Cheshire CW11 0RZ. Tel: (0270) 582748.

Members Corner

Dr Throssell of 4 Bury Lane, Codicote, Hitchin, Herts SG4 8TX (Tel. Stevenage 820266) would like to hear from anyone who can help with implementing double precision arithmetic on the Beeb.

Peter Wilson is having difficulty in driving his Tandy CGP-115 printer/plotter from the serial port of his Beeb and would welcome any help. Peter can be contacted at 156 Ray Lodge Road, Woodford Green, Essex IG8 7PB. Tel. 01-505-7238.

Mr R.W.Sloan is interested in making contact with other BBC users in the Leicester area to exchange ideas. Mr Sloan can be contacted at 12 Bartholomew Street, Highfields, Leicester LE2 1FA.

Robert Armstrong would like to contact anyone who can advise him about BCPL before purchase. Robert is on Marlow 73737.

Events

Electron and BBC Micro User	Royal Horticultural Halls, Westminster	7- 9 Nov
Making the Most of Education	Barbican Centre London	21-24 Jan

←34

Simonsoft,
25 Tatham Road, Abingdon,
Oxon. OX14 1QB

5% to members of BEEBUG on Sprites v.2, Superfruit, Castaway. 5% to members of ELBUG Sprites v.2, Superfruit.

A. Spencer,
74 Dovers Park,
Bathford,
Bath BA1 7UE

25% to members for Journal Indexing System and Objective Test System.

Thurston's Electronics Supplies Ltd.,
Thurston House,
18 Shirehall Lane,
Hendon,
London NW4 2PB

Mail Order Only. Discounts to members on some items. Phone for details.

Total Business Service,
29 Hollow Way Lane,
Amersham, Bucks.

A free program with every two purchased.

Universal Stands,
43 Pearce Avenue,
Parkstone, Poole,
Dorset BH14 8EG
Tel: 0202 740147

5% discount to members on their Universal Stand.

Weserve
128 West Street,
Porchester,
Hants. PO16 9XE

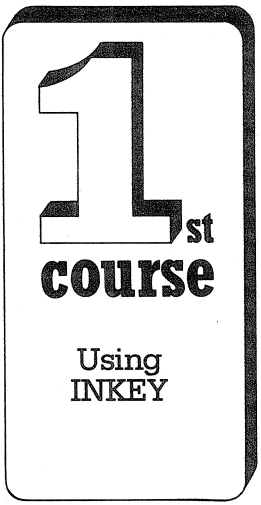
5% to 10% discount on hardware, and 13% off all Rom's/Software.

Wholesale Supplies Ltd.
Woking.
Tel: 04862 62020

Juki 6100 daisywheel printer. Model 6100. Brand new, fully guaranteed, free delivery UK mainland, £369 inc. VAT.

Zero Software,
29 St. Michaels Close,
N. Waltham,
Hampshire RG25 2BP

£1.95 off "EASYCALC" package.



David Graham concludes his look at keyboard input with an exploration of INKEY in its various guises.

In many respects the use of the INKEY and the INKEY$ functions is very similar to that of the GET and GET$ functions treated in the last issue of BEEBUG. In fact, almost anything that will work with GET or GET$ will also work with INKEY or INKEY$. But the INKEY functions are more flexible and can be used in a variety of interesting ways. In addition, there is one particular application of INKEY which can detect machine configuration. This is treated in a separate article in the present issue.

Perhaps the most immediately obvious difference between GET and INKEY is that the latter takes a parameter. Thus to detect a key with GET you might use:

```
A=GET
```

whereas with INKEY the line would take the form:

```
A=INKEY(x)
```

In normal applications, the variable x must be a positive integer between 0 and 32767. Its purpose is to specify the time limit (in hundredths of a second) during which INKEY or INKEY$ will wait for a response. Thus:

```
A$=INKEY$(200)
```

will set A$ equal to the key pressed providing that it is pressed within 2 seconds. Otherwise the function will return with an empty value (A$="", and interestingly, ASC(A$)=-1).

In the case of INKEY, if no key is pressed within the time limit, the value -1 is assigned: otherwise INKEY returns with the ASCII value of the pressed key in just the same way as GET, described in the previous issue. INKEY can therefore be used in the following way:

```
IF INKEY(100)=-1
  THEN PROCkeynotpressed
  ELSE PROCkeypressed
```

The keyboard is monitored for a key press, and if one is made within 1 second then PROCkeypressed is called, and if not, PROCkeynotpressed.

TIME DELAYS

The timeout feature of INKEY and INKEY$ may be put to good use in implementing delay loops in a program. They have the advantage over FOR loops of being able to specify the delay time precisely, and in a way which is machine independent (a useful feature since the Master and any machine with a second processor run more quickly than an ordinary model B). For example:

```
A=INKEY(500)
```

will cause a delay of exactly 5 seconds. It is much more compact than using a REPEAT loop in conjunction with the TIME function, and has the added feature that pressing any key will immediately curtail the delay. This is often desirable, in that a user familiar with a particular program can speed his way through to a given point by abridging INKEY delays with a press of the space bar.

INKEY(0)

Another useful feature of INKEY and INKEY$ is that with the time parameter set to zero, the program can check the keyboard, and continue without any significant delay whatsoever. There are many uses for this. For example, you may have a program which displays a continuously updated clock on the screen, but which also requires keyboard input. The following routine would achieve this:

```
REPEAT
  REPEAT
    PROCclock
    A=INKEY(0)
  UNTIL A>-1
  PROCrespond
UNTIL FALSE
```

The inner REPEAT loop continually updates the screen display using PROCclock, each time checking the keyboard. If any key has been pressed, it drops out of the loop and calls PROCrespond.

This is exactly the method used in Peter Rochford's excellent Master Menu

featured in the last issue. INKEY(0) is used for all keyboard input so that the menu screen can show the on-board analogue clock ticking away the seconds even though the program appears to have halted for keyboard input.

KEY MATCHING

When using the GET or INKEY functions for keyboard input in cases where there is a large range of acceptable inputs, some attention needs to be given to the way in which the responses are handled. If the keys used form a natural ASCII sequence, then this is a relatively easy matter. Using the following, for example:

```
REPEAT
key=INKEY(0)
UNTIL key>lowerlimit AND key<upperlimit
```

the REPEAT loop is only exited when a key is struck within the stated range (i.e. between upperlimit and lowerlimit). A series of individual IF statements could then test for specific values within the range, and direct the program accordingly.

If you have a large number of non-sequential key responses however, other techniques need to be employed. A particularly neat solution to the problem is afforded by the INSTR function, as the following example illustrates:

```
100 ON INSTR("AaBb1*",GET$) GOTO 200,
  200,300,300,400,500 ELSE 100
```

The effect of this statement is to check to see if any of the keys A, a, B, b, 1 or * are pressed. If they are the program is directed to the appropriate line number. If any other key is pressed, line 100 is repeated. As you can see, this particular approach is better suited to GET$ than to INKEY$.

DETECTING THE CURSOR AND FUNCTION KEYS

Neither GET nor INKEY will normally return a value for the Copy, Delete, cursor or function keys. But this is easily remedied with two simple FX calls. FX4,1 allows the cursor, Delete and Copy keys to return ASCII values. They can then be picked up with INKEY, INKEY$, GET or GET$. The ASCII values returned are given in the User Guide. Similarly FX225,128 allows the function keys to return ASCII values. The 128 supplies a so-called offset value for the keys, and in this case means that f0 will return ASCII value 128, f1, 129, and so on. The cursor and function keys should be reset to normal afterwards with FX4,0 and FX225,1 respectively, so as to re-enable cursor key editing and function key use.

NEGATIVE INKEY

INKEY and INKEY$ obtain their response not from the keyboard direct, but, just as with GET and GET$, from the keyboard buffer. For this reason you may wish to flush the keyboard buffer with *FX15,1 before using INKEY or INKEY$ as described in the previous article. However, if you give INKEY a negative parameter in the range -1 to -255, a direct keyboard read will be made. This negative parameter directly replaces the time parameter described above; and INKEY with a negative argument does not wait for a keypress, it merely checks to see if a key is being pressed at the time of the check. Moreover, only one specified key is checked on each occasion, and the result is either TRUE or FALSE. Thus, since the negative INKEY code for the letter "Q" is -17, the following can be used to check if "Q" is being pressed:

```
IF INKEY(-17) THEN PRINT "Q pressed"
```

The response of the negative INKEY function is extremely fast, and you will see that if you enclose this line in a REPEAT loop, every time that you press "Q", "Q Pressed" will be printed several times.

The negative INKEY codes are given in the User Guide, and in our Giant Reference Card. As you will see, unlike the ASCII codes of GET and positive INKEY, they follow no particular pattern. But a second glance reveals a further advantage over the ASCII GET codes: there is a value for every key on the keyboard (except Break). Negative INKEY thus allows you to directly detect whether any of the function keys, the cursor keys, Shift, Ctrl or even Caps Lock is being pressed. One value which is not given is the INKEY code -129. This is a particularly useful one as it tests to see if ANY key is being pressed

Another major advantage of the negative INKEY function over its close relatives is its ability to respond correctly even when a number of keys are held down simultaneously. In such cases, the ordinary INKEY function gives erroneous results, as the short program below illustrates.

```
10 REPEAT:PRINT INKEY$(0);
20 UNTIL FALSE
```

This will print to the screen, any key

pressed. But if you hold down one key, and then press another, it will ignore the first key pressed. And if a third is pressed simultaneously, the keyboard two-key rollover is defeated, and no character is passed through at all.

By contrast, the accompanying program shows how simultaneous detection can be achieved with negative INKEY. When it is run, it continuously monitors the "Z", "X" and "C" keys, and prints them in a central area of the screen if they are being pressed. You will see that now, even if all three keys are pressed together, they are all detected. Tests with the program will show that you can press a great number of other keys simultaneously without causing the "Z", "X", "C" group to read incorrectly; though this can happen. For example, if you press "D", "F" and "R" simultaneously, the program will think that "X" is being pressed. This happens because the keyboard, as with all other micros, is of the polled variety. In other words, a 74 key keyboard does not use 75 individual connections, but gets away with 10 or so. This keyboard multiplexing or sharing of lines can be defeated by holding down numbers of keys simultaneously.

```
10 REM SIMULTANEOUS KEY DETECTION
20 MODE7
30 VDU23,1,0;0;0;0;
40 PRINTTAB(10,4)"Z X C SIMULTANEOUS"
50 PRINTTAB(13,5)"KEY DETECTION"
60 PRINTTAB(14,6)"(Q to QUIT)"
70 REPEAT
80 PRINTTAB(17,10);
90 IF INKEY(-98) PRINT"Z" ELSE PRINT" "
100 PRINTTAB(19,10);
110 IF INKEY(-67) PRINT"X" ELSE PRINT" "
120 PRINTTAB(21,10);
130 IF INKEY(-83) PRINT"C" ELSE PRINT" "
140 UNTIL INKEY(-17)
150 *FX15,1
160 MODE7
```

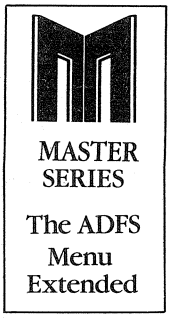
Because the program uses negative INKEY, which reads the keyboard directly rather than taking characters from the keyboard buffer, *FX15,1 has been used at the end of the program to clear the buffer, since otherwise the string of keys pressed would be printed to the screen on exit.

Because of its speed of response and its ability to handle numbers of keys pressed simultaneously, negative INKEY can be used in games programs to check the keyboard for LEFT, RIGHT, FIRE or whatever. A simple splat-em game might adopt the following approach:

```
PROCsetup
REPEAT
  PROCgame
  PROCkeys
UNTIL end
PROCending
END
DEFPROCkeys
IF INKEY(-98) PROCleft
IF INKEY(-67) PROCright
IF INKEY(-74) PROCfire
ENDPROC
```

Here the game is played within a REPEAT loop, accessing PROCgame and then PROCkeys. This latter just checks for "Z", "X" or Return, responding with PROCleft, PROCright or PROCfire respectively. Using this structure, the program will respond correctly even if more than one key is left pressed down. For example, holding down "Z" and Return will cause both PROCleft and PROCfire to be called repeatedly.

The negative INKEY function can therefore be an extremely useful one to use at times. But remember that each statement can only check for a single key, so that if you need to check for many, you are probably better off with GET or the normal positive INKEY function. Generally speaking GET and GET$ should be used when there is no time limit on the key checking - or when the program must stop for a keyboard response. INKEY or INKEY$ come into their own when time plays a part, and the program must perform other functions while appearing to simultaneously check the keys, or where a keyboard response is required within a certain time limit. Finally the negative INKEY function is used when you wish to detect keys which cannot be detected with a straightforward GET or INKEY, or where speed of response, and response to simultaneous keypresses is important.



MASTER SERIES

The ADFS Menu Extended

David Graham shows how to extend Peter Rochford's ADFS disc menu, featured in the last issue, enabling it among other things to auto-load View and Wordwise files.

The purpose of this article is to provide some enhancements to Peter Rochford's excellent ADFS auto-menu featured in the last issue. The modifications are grouped into two parts: a number of minor adjustments are followed by more detailed customising for Wordwise Plus and View, including a routine to read the date from the Master's calendar into Wordwise. In all cases it is assumed that the line numbering of the program as published in the magazine has not been altered.

MINOR MODS

To improve the error handling (our oversight, not the author's) insert line 915, and alter line 920 as follows:

```
915 MODE 128
920 REPORT:PRINT" at line ";ERL:PRINT
  '"Press Space":IF GET THEN RUN
```

Altering line 350 allows the menu to display ALL files in a given directory rather than just those with the appropriate ACCESS status. To achieve this delete the following from that line:

```
IF((C%?&E EOR2)AND3)=3
```

To make the cursor-down key take you to the parent directory (as distinct from cursor-up, which takes you to the root directory), insert the following line:

```
185 IFF%=1380SCLI"DIR ^":PROCH
```

If you wish to change the colour of the display, alter the VDU 19 call in line 310. Using VDU19,1,6| will give cyan on black for example.

If you would prefer to see the display build up rather than wait in darkness at the beginning of the program, remove the following from the start of line 150:

```
VDU19,1,0|:
```

WORDWISE PLUS EXTENSIONS

The purpose of the following modifications is to allow Wordwise files to be selected from the ADFS menu, and be automatically loaded into Wordwise Plus on selection, and for the date to be transferred across from the Master's calendar into the Wordwise Plus variable D$. It can then be placed anywhere in a document with the embedded command: PS D$. This is achieved by creating a file named "$.startup" directly from within the menu program, and then using *EXEC to load the file back in. This calls Wordwise and transfers the date data as well as loading in the selected text file. The filename is transferred to two Wordwise variables F$ (which temporarily holds the current filename), and A$ which will retain the loaded filename until that variable is reset by the user. The routine also sets up the function keys, though this can be left out if not required.

To achieve these enhancements, you need to insert a single line into the menu program as follows:

```
245 IF C%?7=&FF THEN GOTO 1000
```

Then append the accompanying listing to the menu program.

VIEW EXTENSIONS

The View customising process is very similar. The program is a little shorter because no date details are transferred across. This is unnecessary since View can read the date automatically using the |D option (see accompanying Master Hints). But if you want to transfer other data into View, you could use the same principles used in the Wordwise enhancement to create a View macro file.

To achieve the View customising, insert the following line into the menu program, and append the accompanying listing:

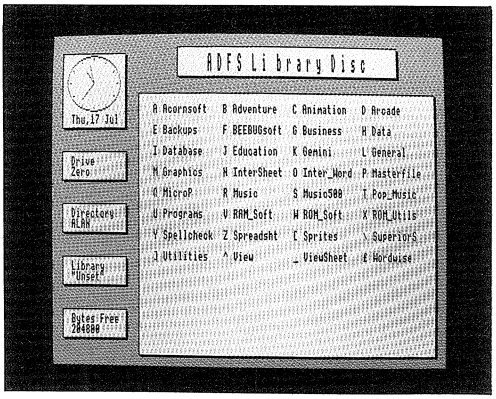
```
242 IF LEFT$(A$(F%-64),1)="v" THEN
  GOTO 2000
```

Unfortunately, View load and execution addresses cannot be used as an infallible indication of a View file, and we have had to resort to using the filename itself. If these amendments are used, the menu will assume that all files with names starting with lower case "v" are View files.

As with the Wordwise amendments, a routine is included to set up the function

keys, and this may be omitted if it is not required. The Wordwise and View modifications have been given separately, but both may be added to the same menu if you wish. If you do this, there is a single procedure definition which is used by both routines, and which need not be duplicated. This is DEF PROCwrite.

This month's magazine cassette/disc contains a full copy of the extended menu. This covers the Wordwise, View, parent directory, display all files, and error trapping enhancements.




```
1000 REM WORDWISE ADDON
1010 R$=A$(F%-64)
1020 ZZ%=OPENOUT("$.startup")
1030 PROCdate
1040 PROCwrite("*WORDWISE")
1050 PROCwrite(":D$="+CHR$34+wwd$+CHR$
34)
1060 PROCwrite(":LOAD TEXT "+CHR$34+R$
+CHR$34)
1070 PROCwrite(":A$="+CHR$34+R$+CHR$34 )
1080 PROCwrite(":F$=A$"+CHR$13)
1090 CLOSE# ZZ%
1100 PROCkeys
1110 *EXEC $.startup
1120 END
1130 :
1140 DEFPROCdate
1150 day$=MID$(TIME$,5,2)
1160 month$=MID$(TIME$,8,3)
1170 year$=MID$(TIME$,12,4)
1180 IFLEFT$(day$,1)="0" THEN day$=" "
+RIGHT$(day$,1)
1190 wwd$=day$+" "+month$+" "+year$
1200 ENDPROC
1210 :
1220 DEFPROCwrite(string$)
1230 FORA=1TOLEN(string$)
1240 BPUT# ZZ%,ASC(MID$(string$,A,1))
1250 NEXT
1260 BPUT# ZZ%,13
1270 ENDPROC
1280 :
1290 DEFPROCkeys
1300 REM SET KEYS HERE e.g.
1310 *KEY 0 etc.
1320 ENDPROC
```

```
2000 REM VIEW ADDON
2010 R$=A$(F%-64)
2020 ZZ%=OPENOUT("$.startup")
2030 MODE 128
2040 PROCwrite("*WORD")
2050 PROCwrite("LOAD "+R$)
2060 CLOSE# ZZ%
2070 PROCkeys
2080 *EXEC $.startup
2090 END
2100 :
2110 DEFPROCwrite(string$)
2120 FORA=1TOLEN(string$)
2130 BPUT# ZZ%,ASC(MID$(string$,A,1))
2140 NEXT
2150 BPUT# ZZ%,13
2160 ENDPROC
2170 :
2180 DEFPROCkeys
2190 REM SET KEYS HERE e.g.
2200 *KEY 0 etc.
2210 ENDPROC
```

ADFS MENU ERRATA

We regret that in the process of listing the ADFS Menu program, the six occurrences of the vertical bar character "|" failed to be printed. Fortunately the program still works in this state, though the display is not completely as expected.

To remedy the problem, insert a vertical bar character in place of the space printed at the following points:

Line 110 between BOOT and M
Line 120 between VDU23,1 and :
Line 150 between VDU19,1,0 and :
Line 310 between VDU19,1,7 and :
Line 870 between VDU23,1,1 and :
Line 890 between VDU23,1 and :

CASSETTE/DISC VERSION

This does not suffer from the loss of characters mentioned above, but it is also not identical to the version published in the magazine. In particular the star command option is not implemented, and the line numbering is different. For this reason, the extensions treated in this issue relate to the version printed in last month's magazine.

David Graham presents a round-up of hints, tips and ideas for the Master.

ONE LINE CLOCK/CALENDAR

The following single liner gives you an instant mode 7 double-height clock and calendar:

```
10 MODE7:VDU23,1|:REPEAT:FORA=1TO2:PRI
NTTAB(4,9+A)CHR$131;CHR$141;TIME$:NEXT:U
NTIL 0
```

MULTIPLE PARENTS

The ADFS allows the "^" symbol to be used to mean "parent directory". Parents can be stacked as follows:

```
*. ^.^.^.^
*DIR ^.^.^
```

The first command catalogues the directory which is four higher than the current directory. The second sets the directory to three higher than currently.

NOTES ON *MOVE

This extremely useful command will copy (rather than move) files across filing systems, and therefore is a must for ADFS users, since it will allow you to transfer files between DFS and ADFS discs with ease. The various Master manuals however give very little information about it, so here is an example of its use:

```
*MOVE -DISC-:1.W.text -ADFS-:0.$.BACK
  UP.text1
```

This will copy the file W.text on DFS format in drive 1 on to an ADFS disc in drive 0, placing the new file in directory BACKUP, and giving it the name text1. If you have previously declared drive 1 under DFS and performed a *MOUNT 0 under the ADFS, then you do not need to specify the drives; or indeed the directories if these are the current directories. The command would then reduce to:

```
*MOVE -DISC-text -ADFS-text1
```

If the copy operation takes place within a given filing system, it is no longer necessary to specify the filing system, and the syntax is exactly as that for *COPY on the ADFS, except that a filename must be supplied for the destination, because unlike *COPY, *MOVE allows you to change filenames when it copies.

There is a slight quirk with *MOVE from the point of view of the access status of files. If you transfer files between the two disc filing systems, access status is altered. A file with WR status in the ADFS becomes locked when copied to the DFS, and an unlocked DFS file becomes locked to read but not to write operations, though no access parameters appear in the catalogue. In both cases, *ACCESS can be used to reinstate the desired status.

There is another feature of *MOVE which distinguishes it from *COPY. It does not overwrite user memory. This is because it uses the shadow memory area as workspace - all very clever stuff. Because of this you should avoid calling *MOVE when high resolution shadow memory is in use.

CONFIGURING FILING SYSTEM AND LANGUAGE

To configure the ADFS as the default filing system on power-up, use:

```
*CONFIGURE FILE 13
```

This is because 13 is the ROM number of the ADFS ROM (use *ROMS to check this). To revert to the DFS on power-up, use:

```
*CONFIGURE FILE 9
```

and remember that you must press Break to initialise the change.

Similarly, to configure the language to be entered on power-up, use:

*CONFIGURE LANG n

where n is the ROM number of the required language, as follows:

View	14
Basic	12
Editor	11
Viewsheet	10

SHORTER VDU CALLS

The vertical bar character "|" (found to the left of the left cursor key) may be used to terminate long VDU calls which normally end in a series of zeros. This character is very conveniently taken to mean "as many zeros as it takes to satisfy the syntax requirement". Thus the cursor on/off commands become:

```
Cursor off  VDU23,1|
Cursor on   VDU23,1,1|
```

This is far more memorable than the

commands in their original state. It also simplifies the VDU19 colour change commands.

FUNCTION KEYS IN THE EDITOR

The editor instructions explain that the user defined function keys are only active under very limited circumstances. This restriction can be avoided with the following FX call executed either from within or outside the editor:

```
*FX228,1
```

Once the call has been made, Shift-Ctrl with any of the function keys can be used when entering text to supply frequently used words or strings of words. Moreover the keys may now also be used to supply data for the various editor commands. So for example, if you set up key 4 as follows:

```
*KEY4 BASIC|M
```

You can exit the editor to Basic by pressing Shift-f4 to "Return Language" then Shift-Ctrl-f4 to supply the word Basic. Similarly you can supply filenames for saving or loading, or a frequently used line number to be used in response to f0 "Go To Line".

VIEW DATING

View contains a facility to automatically print the date from the Master's calendar. To use it you need to place the following two characters in your text:

```
|D
```

They must follow one of the format commands DH, DF, RJ, LJ or CE.

LIST IF SEARCH

The LIST IF instruction provides a simple search facility which may be used as an alternative to the comprehensive search options provided by the editor. To search for all occurrences of PROCdate in a program, type:

```
LIST IF PROCdate
```

Remember that the search is case-specific so that occurrences of PROCDATE will not be listed. See the Reference Manual Part Two for further details.

USING TIME$

As most users will be aware, the variable TIME$ can be used both to read and write to the Master's internal clock/calendar. Sometimes it is useful to extract specific parts of the time and date information. The following table should assist in this:

The Date	MID$(TIME$,5,11)
The Day (words)	LEFT$(TIME$,3)
The Day	MID$(TIME$,5,2)
The Month	MID$(TIME$,8,3)
The Year	MID$(TIME$,12,4)
The Time	RIGHT$(TIME$,8)
Seconds	RIGHT$(TIME$,2)
Minutes	MID$(TIME$,20,2)
Hours	MID$(TIME$,17,2)

ON PROC

The Master's version IV Basic has a useful enhancement to the ON ... statement. It allows procedures to be called as follows:

```
100 ON X PROCone,PROCtwo,PROCthree
```

When line 100 is encountered, it will call PROCone if the value of X is 1, PROCtwo if it is 2, and so on. Further details appear in section K.1-1 of Reference Manual Part Two.

OBSERVING SHADOWS

Two useful FX calls allow you to write to and display shadow and main screen memory without performing a mode change. This instantly provides a dual screen capability, which can be used both for animation and notepad applications. Here are the calls.

```
*FX112,0  Write to current screen
*FX112,1  Write to main memory
*FX112,2  Write to shadow memory
*FX113,0  Display current screen
*FX113,1  Display main memory
*FX113,2  Display shadow memory
```

FX113 can thus be used to switch instantaneously between two screens. If they are of the same mode, this could be used to achieve certain animation effects. Alternatively the technique could be used to allow the build-up of complex screens, while the user is presented with an entirely different screen which may even be in a different mode. FX112 allows the user to choose which screen he writes to, irrespective of which of the two screens are being displayed.

PRINTER SURVEY

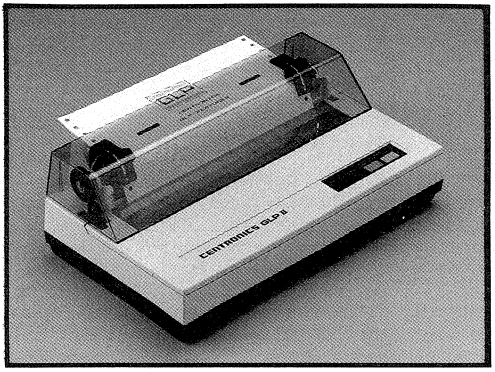
Geoff Bains presents our latest survey of dot-matrix printers for the Beeb market.

Most dot matrix printers support near letter quality printing. Such printers are ideal for many BBC micro users. In 'draft mode' they provide fast printing, useful for producing program listings and other draft material. In NLQ mode all of the printers looked at here are hard to distinguish from the 'real thing' - daisy wheel or typewritten print.

Although NLQ printers are commonplace, the range of facilities, quality and prices is as large as ever. These printers vary from the cheap to those costing more than your Beeb.

CENTRONICS GLP II

The Centronics GLP II is the cheapest NLQ printer around. It is also small - only about 12 x 6in. The GLP has no tractor feed as standard but one can be added for £10.



Despite its price, the GLP can produce all the effects, such as underline, italics, super and subscript, as the more expensive models. However, it cannot produce italic print in NLQ mode.

Usefully, the NLQ mode can be selected from a switch on the outside of the printer. For this price the NLQ print is surprisingly good. It is not all that fast (even slower than the lethargic Epson LX-80) and it can be a little blotchy but for £195 you can hardly complain. The draft mode printing is faint but it's still quite legible and so serves its purpose. The GLP II is undoubtedly a bargain.

```
The very cheap Centronics
  GLP II can print quite
 reasonably in NLQ mode.
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
        1234567890
```

CITIZEN MSP-10E

This is the latest version of the Citizen MSP range. The major improvement over the MSP-10 is a dramatic increase in speed. This printer is much faster than the likes of the Epson LX-80.

However, there is little else that can really be said in favour of this machine. It is a little flimsy and very noisy. Although the MSP-10E has all the features of other machines such as the Taxan KP-810 it costs considerably more. If speed is what you need, the new Taxan KP-810PC+ is a much better bet.

```
The Citizen  MSP-10E looks
 rather like an old Epson
 but performs quite well.
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
        1234567890
```

STAR NL-10

This is a remarkable machine. It is the latest model from Star and is a vast improvement of the previous down market models. It is excellently made and looks good too. It has an automatic paper load facility for cut sheet paper that makes loading single sheets very quick and

simple. If you use a lot of cut sheets then an automatic sheet feeder to take single sheets of paper in quantity is available for £63.

Even in draft mode the NL-10's print is good. In NLQ mode it is excellent. The characters are crisp, clear, and dark. They are a little taller than most but this just adds to the clarity.

The form/line feed switches on the front of the printer can be used in combination to select most functions of the printer. The NL-10 can produce enlarged text that is not only enlarged width-wise but vertically too. There are three sizes of enlargement - the biggest is around half an inch high - ideal for banners and headlines.

The NL-10 comes with a separate interface cartridge that plugs in at the back of the machine. This cartridge not only holds the interface electronics but much of the 'intelligence' of the printer too. The cartridge tested was a Centronics interface with Epson compatible printer functions, suitable for the Beeb. A serial interface is also available for £79 and others for IBM, Commodore and Apple (if the printer is to be shared with another micro) cost £39.

For £320 (including one parallel interface cartridge) the Star NL-10 costs more than the Taxan KP-810. However, like the Taxan, it is starting to find discounted outlets and has been on offer recently for just £260. Even at £320 the NL-10 is hard to beat.

The Star NL-10 is not expensive but gives an excellent quality of print
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

TAXAN KP-810PC+

This new version of the old KP-810 is really intended for users of the IBM PC. It has had its control and character set ROMs replaced by Taxan to give it IBM compatibility and many extra features.

The character set is now IBM PC compatible which means there are 255 characters available. The foreign accented characters are not selected in alternative groups, as the Epson standard, but all are always available. The £ sign, for example is character 156. The IBM standard also calls for a lot less in the way of control codes but none of the Epson ones have been sacrificed in the KP-810PC+.

The quality of the print in draft and NLQ mode is pretty much the same as that of the older model. However, the machine is now faster, giving a claimed 160 cps in draft mode (the same as the MSP-10E), and has a more accurate dot alignment, giving neater graphics. The NLQ mode can now be selected from the front switch panel. Most importantly the double height, condensed, superscript and subscript, and double strike printing are all now available in NLQ mode. NLQ italics are also available if a separate italic character set ROM is fitted (about £50).

The Taxan KP-810PC+ is an improved version of the old favourite.
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

If you are not worried about the loss of true Epson compatibility (and the loss is not great) the KP-810PC+ is a fine machine with an excellent pedigree.

Printer	Price (RRP)	Draft speed	NLQ speed
Centronics GLP II	£195	48	183
Citizen MSP-10E	£454	30	106
Star NL-10	£320	34	155
Taxan KP-810PC+	£349	30	108

Speeds are the times in seconds taken to print a full page of text - the smaller the better.

NOTE: Since this information was prepared the Panasonic KX-P1080 printer has appeared at £189.75 to rival the Centronics GLP II reviewed above. This latest printer also appears to offer excellent value at the price.

RAMROD

RAMROD is the latest all-purpose utility ROM from Clares, covering the Tube, DFS and ADFS, and sideways RAM,as well as all the more obvious features of the Beeb. Geoff Bains gives his assessment.

Product: Ramrod
Supplier: Clares Micro Supplies, 98 Middlewich Road, Rudheath, Northwich, Cheshire CW9 7DA. (0606) 48511
Price: £40 (ROM) £35 (Disc)

Ramrod is a 16K ROM that provides 54 commands to help you use a model B, B+, or Master 128. Ramrod is supplied either as an actual ROM or as a ROM image on disc to load into sideways RAM. Either way, Ramrod is an ingenious collection of utilities covering a wide range of requirements.

The functions of Ramrod fall into four categories - General, Tube, Disc and ROMs. Indeed, there are so many commands that no complete *HELP list is provided. Instead, the display of the commands' syntax is split into four, accessed with an extended *HELP command such as *HELP GENERAL.

Most Ramrod commands are used to manipulate memory. The General section has commands to search for bytes or strings, to edit memory and to disassemble machine code programs. These are given the names BYTE, FIND, EDIT and DISS, prefixed with 'M' for main memory area or 'S' for the shadow memory area, if present.

Numeric parameters to specify start and finish addresses and so on can, where relevant, be specified in either hex or decimal by preceding the characters with the usual '&' for hexadecimal or '/' for decimal. However, default number bases for each parameter means that you don't have to trouble about this. Parameters can even take the value of a resident integer variable (A% - Z%) simply by giving the variables name in place of the parameter.

All four commands are based around the same routine. EDIT, FIND, and BYTE produce a mode 7 display of the section of memory in dump format. If a search command initiated the display, the first find is shown at the top of the screen, and the Tab key moves on to the next. DISS gives a display with disassembled mnemonics. Whichever command was used, Ctrl-Tab switches between dump and disassembler format. Both displays are also dynamic, changing as memory contents change.

The whole display can be scrolled back and forth with the cursor keys and the memory altered in either hex or ASCII. The current ROM is displayed at the top of the screen and can be altered to inspect or disassemble another sideways area (or even edit it if it's in sideways RAM). The disassembler is very fast. Indeed, the whole routine is easy to use and provides a very effective series of facilities.

As well as the scrolling disassembler and editor, simpler versions are also provided. *MNEMONIC, disassembles from a specified address in a mode 0 format. *TEXT and *HEX dump memory in in hex or ASCII, again in a mode 0 format.

Other memory utilities are included. *FILL seeds an area of memory with a specified byte. *SHIFT moves an area of memory to a new address. *RELOCATE changes a piece of machine code to run at a different location. This isn't fool proof; and no relocation program can be. It must be used with a reasonable knowledge of the machine code program in question.

Such commands can act on sideways ROM or RAM, as well as main memory. Ramrod has a clever way of specifying which sideways ROM area is to be used. If the memory to be used is in the normal RAM area then start and finish addresses are specified as normal. A sideways ROM is indicated by designating the start address as &8000 and adding the ROM name or socket number onto the end of the command. In this way a common command structure suits all the eventualities.

The General section covers other functions too. *VECTOR prints a list of the current settings of all the page two vectors. *CHECK verifies a Basic or data file on cassette or disc with one in memory. *MCOMPARE does the same between

two areas of memory, including sideways RAM/ROM. Two more familiar utilities, but with a new twist, are *SKEY and *LKEY. These save and load function key definitions from tape or disc. The twist is that they cope with the Master 128 which stores the definition in its inaccessible private workspace.

The prefixing of the commands is used a lot in Ramrod. Many of the General commands are used in the Tube section, prefixed with a 'T'. This leads to rather unpronounceable (and therefore unmemorable) commands, but does mean that a consistent nomenclature and syntax is used right across Ramrod's wide range of facilities.

Four extra Tube commands are included. *IOTCOMPARE will compare an area of memory in the IO processor with one across the Tube. *TIOCOMPARE performs the reverse. For those interested in using the Tube for more than just running commercial software, two useful commands are *IOTSHIFT and *TIOSHIFT. These transfer data between the two processors, previously a very complex process.

The Disc section of Ramrod also offers the three operations - BYTE, FIND, EDIT. For discs these commands are prefixed with a 'D' for a normal DFS disc (including the 1770 DFS used on the Master) or an 'A' for ADFS users. These commands produce the same scrolling dump display as used for the memory and Tube commands.

Naturally, operation is slower as disc access is required, but Ramrod provides one of the best disc sector editors around. An 'M' can be added as a parameter following the command to specify a mode 0 display instead of the usual mode 7. If 80 column output is chosen in this way, but the machine has no shadow memory, then a 'Go (Y/N)' prompt is given before the command is executed, as a protection against corruption of important data that maybe overwritten. It's a shame this option is not provided for the memory commands.

The final section of Ramrod is concerned with sideways ROM and RAM. This provides one of the best collection of ROM-managing utilities around. *RON and *ROFF enable and disable a ROM and a disabled ROM can be kept off through Ctrl-Break. *RWIPE clears a bank of sideways RAM and *RLIST lists all the ROM/RAMs in the machine along with their type and enabled state. ROM images can be loaded from disc with *RLOAD and moved or swapped between ROM sockets with *RMOVE and *RSWAP (as long as they are suitably equipped with sideways RAM).

Perhaps the most useful command is *RPASS, used to pass a command directly to a named ROM and so avoid any command conflicts.

Ramrod also provides several utilities to use the ROM filing system. *RLOAD will format a bank of sideways RAM ready for program storage. *RTITLE will name it and *RINFO detail the programs there. *RFILE is used to add any Basic or machine code program to the bank.

There is nothing very new about many of these facilities. You'd be forgiven for thinking Ramrod is just another forgettable utility ROM. Although Ramrod's features are mostly familiar they are implemented more efficiently than most of their predecessors. However, Ramrod's real appeal lies in the sheer number of utilities provided with a common syntax and its compatibility across the whole range of BBC micros.

There are few ROMs around that cover as much ground as Ramrod. If you already have a number of the utilities provided by Ramrod then £40 is going to be a lot to pay for a few more. However, if your ROM sockets are not yet fully stocked, or if you want to rationalise your collection, Ramrod is hard to fault.

POINTS ARISING POINTS ARISING POINTS ARISING POINTS

BEEBUG FILER (BEEBUG Vol.5 Nos.2 & 4)

In the Filer Graphics program (magazine cassette/disc only) edit line 3320 to read:

```
3320 FORI=1TOf:INPUT#F,p$,p%:field$(I)=FNstrip(p$,"."):NEXT
```

In the filer Accounts program change the value of 512, assigned to FDR in line 1060, to 256 for compatibility with the original Filer program. This is incorporated in this month's listing, and in the complete Accounts program on the magazine cassette/disc.

BEEBUG WORKSHOP - DATE HANDLERS (Part 1)

Dates can be awkward things to handle in computer programs. Surac takes a detailed look at the problems and their solution, with a comprehensive set of routines.

Dates occur in a variety of formats, and with the different number of days in each month, can prove a nuisance to handle correctly in programs. This month's Workshop describes a comprehensive set of routines designed to make this task very much easier. These routines will input dates, and check them; they will manipulate dates by adding or subtracting days, whole weeks, months or years; and they will format a date into a string which may take any form from "12/11/85" to "Wednesday 12 November 1985" and beyond.

All the routines are structured to allow their use within any BBC Basic program. The only global variables are some arrays containing the numbers of days in the month and the names of months and days of the week, and three integers global to some very 'deep' sub-routines, which you are unlikely to need on their own.

For those readers who wish to use the software, without necessarily going into the whys and wherefores, simply type in the routines you need, and merge them with your existing program. Just make sure you provide the correct parameters when using the routines.

DATE INPUT/VALIDATION

This routine expects input using a format of DD/MM/YY, DD/MM/YYYY or simply DD/MM. If a two digit year is entered then it is assumed to be preceded by '19', i.e. '64' would be stored as 1964. If the year is missed off altogether it is assumed to be 1986 (see line 1080). These features can save a lot of typing for harassed secretaries.

```
 1000 DEFFNdatein(p$):LOCALdate%:date%=F
Ndateps(FNstrnck(p$+" (D/M/.)",3,11,FALS
E)):IFdate%THEN=date%ELSEPROCbipa:=FNdat
ein("WHAT?")
 1010 DEFFNdateps(date$):LOCALyear%,mont
h%,day%:year%=FNyearis(date$):month%=FNm
onis(date$):day%=FNdayis(date$):IFyear%<
1=FALSE ELSEIFyear%<100year%=year%+1900
 1020 =FNdatepi(day%,month%,year%)
 1030 DEFFNdatepi(d%,m%,y%):PROCfeb(y%):
IFm%<0ORm%>11ORy%<1752ORy%>9999ORd%<0=FA
LSE
 1040 IFd%>mon%(m%)=FALSE ELSE=d%+m%*&10
0+(y%MOD100)*&10000+(y%DIV100)*&1000000
 1050 DEFFNlpyr(year%):IFyear%MOD4<>0=FA
LSE ELSEIFyear%MOD400=0=TRUE ELSE IFyear
%MOD100=0=FALSE ELSE=TRUE
 1060 DEFPROCfeb(y%):IFFNlpyr(y%)mon%(1)
=29ELSEmon%(1)=28
 1070 ENDPROC
 1080 DEFFNyearis(date$):LOCALa%:a%=INST
R(date$,"/",1):IFa%<1THEN=FALSE ELSEa%=I
NSTR(date$,"/",a%+1):IFa%<1=1986ELSE=VAL
(RIGHT$(date$,LEN(date$)-a%))
 1090 DEFFNmonis(date$):LOCALa%,b%:a%=IN
STR(date$,"/",1):IFa%<1=FALSE ELSEb%=INS
TR(date$,"/",a%+1):IFb%THEN=VAL(MID$(dat
e$,a%+1,b%-a%-1))-1ELSE=VAL(RIGHT$(date$
,LEN(date$)-a%))-1
 1100 DEFFNdayis(date$)=VAL(LEFT$(date$,
INSTR(date$,"/",1)))
 1110 DEFPROCbipa:SOUND1,-15,10,10:ENDPR
OC
 1120 DEFFNstrnck(p$,min%,max%,flag%):LO
CALb$:@%=1:PRINTp$;:IFflag%GOSUB1170
 1130 INPUT"? "b$
 1140 IFLENb$>max%PROCbipa:=FNstrnck("TO
O LONG?",min%,max%,TRUE)
 1150 IFLENb$<min%PROCbipa:=FNstrnck("TO
O SHORT?",min%,max%,TRUE)
 1160 =b$
 1170 PRINT" (MAX "max%;:IFflag%THENPRIN
T" MIN "min%;
 1180 PRINT")";:RETURN
```

If you enter a date which does not exist for any reason, e.g. 29/2/86 or 31/9/95, the program responds with WHAT??

and requires another input; similarly if you enter any other character apart from a number or a slash.

TECHNICAL NOTES

The above routine is in the form of a function (FNdatein) which returns an integer representing the date. This integer number is compatible with any other routines requiring the date as a number, and does not need any further processing before being manipulated or formatted. The four bytes of an integer are used as follows: the most significant byte holds the century, the next most significant the parts of a century (i.e. the tens and units of years), the next-to-least significant the months, and the least significant the days. The months and days start at zero rather than one, to allow direct manipulation and referencing of arrays. Within each byte, the number is expressed in binary. Further compression would have been possible, but more awkward than the space saved would justify. The dates may be directly compared, and are easily sorted.

SUMMARY OF FUNCTIONS AND PROCEDURES USED

FNdatein(prompt$) returns an integer containing the date in packed form. This routine is recursive (i.e. calls itself) until a valid date is entered. The parameter supplied is a prompt, with correct punctuation etc added automatically.

FNdateps(date$) converts a date in string form to packed integer. 1900 is added to years between 1 and 99 here.

FNdatepi(day%,month%,year%) converts three integers (day, month and year respectively) into a single packed integer date.

FNlpyr(year%) returns TRUE if the integer passed to it is a leap year, otherwise FALSE. It obeys the 400 year rule.

PROCfeb(year%) accepts the year as an integer parameter, and alters the value of mon%(1) from 28 to 29 if a leap year.

FNyearis(date$) accepts the date as a string, and returns an integer year. The 1986 assumption is made in this routine.

FNmonis(date$) and FNdayis(date$) chop the input string and return integers in like manner to FNyearis.

PROCbipa makes a distinctive low tone to signal errors etc.

FNstrnck(prompt$,minlen%,maxlen%,flag%) accepts four parameters and returns a string which has been typed in. More than just a 'mugtrap', it supplies a prompt and other information to the operator. The last parameter is a flag set TRUE to cause display of the length limits as part of the prompt. FNstrnck is recursive until a valid string is input.

```
 1200 DEFFNdayip(d%)=d%AND&FF
 1210 DEFFNmonip(d%)=(d%AND&FF00)DIV&100
 1220 DEFFNyearip(d%)=((d%AND&FF000000)D
IV&1000000)*100+(d%AND&FF0000)DIV&10000
 1230 DEFFNupday(dt%,nd%):LOCALd%,m%,y%:
PROCdmyout:d%=d%+nd%:ONSGN(nd%)+2GOSUB12
40,1270,1280:PROCyok:=FNdatepi(d%,m%,y%)
 1240 IFd%>0RETURN
 1250 REPEAT:m%=m%-1:IFm%<0m%=11:y%=y%-1
:PROCfeb(y%)
 1260 d%=d%+mon%(m%):UNTILd%>0:RETURN
 1270 RETURN
 1280 IFd%<=mon%(m%)RETURN
 1290 REPEAT:d%=d%-mon%(m%):m%=m%+1:IFm%
>11m%=0:y%=y%+1:PROCfeb(y%)
 1300 UNTILd%<=mon%(m%):RETURN
 1310 DEFFNupmon(dt%,nm%):LOCALd%,m%,y%:
PROCdmyout:m%=m%+nm%:ONSGN(nm%)+2GOSUB13
20,1340,1350:PROCyok:=FNdatepi(d%,m%,y%)
 1320 IFm%>-1RETURN
 1330 REPEAT:m%=m%+12:y%=y%-1:UNTILm%>-1
:RETURN
 1340 RETURN
 1350 IFm%<12RETURN
 1360 REPEAT:m%=m%-12:y%=y%+1:UNTILm%<12
:RETURN
 1370 DEFFNupyear(dt%,ny%):LOCALy%,m%,d%
:PROCdmyout:y%=y%+ny%:PROCyok:=FNdatepi(
d%,m%,y%)
 1380 DEFPROCdmyout:y%=FNyearip(dt%):m%=
FNmonip(dt%):d%=FNdayip(dt%):PROCfeb(y%)
:ENDPROC
 1390 DEFPROCyok:IFy%>9999y%=9999:d%=31:
m%=11:PROCbipa
 1400 IFy%<1752y%=1752:d%=1:m%=0:PROCbip
a
 1410 PROCfeb(y%):IFd%>mon%(m%)d%=mon%(m
%)
 1420 ENDPROC
```

THE DATE MANIPULATION SET

These functions and procedures can be used individually for various operations on data held in the standard integer format already described. They are very useful for stepping through diaries or manipulating dates generally. There is a whole set of these, with fairly obvious

→5

Dynamically Called Functions

David Graham shows how to call functions dynamically, and points the way to writing more efficient code.

In the Workshop for BEEBUG Vol.5 No.3 a routine was evolved for calling procedures dynamically by variable name. This is a highly desirable thing to be able to do, and has been partly implemented on the Master series with the new ON-PROC construct. But the method adopted in July's Workshop used somewhat sneaky code, poking characters into the Basic program area to achieve its ends. There is in fact a much simpler, and totally clean method, but using functions.

It makes use of EVAL (see First Course in BEEBUG Vol.4 Nos. 8 & 9 for more information on this) to call a function rather than a procedure, but the effect is identical. Thus the following statement will call a function whose name is held in the variable A$:

```
dummy=EVAL("FN"+A$)
```

If A$="setup" then the statement will cause FNsetup to be called. As you can see, the function returns a dummy value, and is thus equivalent to a procedure. It could of course be used to return a useful value if this was required.

There are many applications for such a useful construct. For example, the following line can be used in place of the Master's ON-PROC structure:

```
dummy=EVAL("FNtest"+CHR$(A))
```

The effect of this would be to call FNtest1 if A=1, FNtest2 if A=2, and so on.

This technique also provides an extremely efficient way of handling the input from a menu, as the accompanying program illustrates. This looks for keyboard input of any of the following characters A, a, B, b, C, c, 1, 2, 3 or *. The code in line 60 ensures that it accepts both upper and lower case letters. It then calls the appropriate function. If "A" or "a" is pressed it calls FNmenuA and so on. Only two of the functions have been defined in this illustration, so that pressing "B" for example will call a non-existent function. However, all keys except for the 10 nominated are locked out by the conditions set up in line 70, and the program only requires the addition of the five remaining function definitions to make it complete.

The example below is coded to return a dummy value of 0. Any value would suffice as it is there just to satisfy the syntax of a function definition.

```
 10 REM DYNAMIC FUNCTION MENU
 20 :
 30 REPEAT
 40 :
 50 REPEAT
 60 A=GET:A$=CHR$(A+32*(A>96))
 70 UNTIL INSTR("ABC123*",A$)
 80 :
 90 DUMMY=EVAL("FNmenu"+A$)
100 UNTIL FALSE
110 :
120 DEFFNmenuA
130 PRINT"A selected"
140 =0
150 :
160 DEFFNmenu1
170 PRINT"1 selected"
180 =0
```

1 functions. The full set of routines consists of a day adder, a month adder, a year adder, and a 'days between dates' function.

TECHNICAL NOTES

The first three functions return the day, month or year number as a simple integer from the the date (packed date to integer). FNupday(date%,daystogo%) requires two parameters; the date as a packed integer, and the number of days to be added (this last can be zero or negative). It returns the date as modified, in packed integer form.

FNupmon(date%,monthstogo%) and FNupyear (date%,yearstogo%) do the same as FNupday, for the months and years. When stepping by months from the end of a month, FNupmon will not go beyond a month end; e.g. one month from 31 January gives the last day of February. [To be continued next month.]

Filer Accounts Option (Part 2)

Mike Williams concludes the accounts option for the Filer database system by adding the code to handle direct debits and standing orders.

This month, we will add the additional code to the Filer Accounts program given last month, so that it will automatically handle direct debits and standing orders, in addition to all your other payments.

ADDING THE NEW CODE

The additional code is listed at the end of this article and this must now be added to the original Accounts program. Note that in some cases new instructions replace others used previously. To add the new code proceed as follows:

1. Enter the new instructions, and save separately as a safeguard. E.g.:
   SAVE"FILERT" <Return>

2. Now create a spooled version of this code, for example:
   *SPOOL FILERX <Return>
   LIST <Return>
   *SPOOL <Return>

3. Load in the original Accounts program and append (using *EXEC) the new spooled code to it, thus:
   LOAD"FILERA" <Return>
   *EXEC FILERX <Return>
   (ignore any error messages)
   SAVE"FILERB" <Return>

4. Once you have tested the extended program, called FILERB above, you can re-save it as FILERA (as your main accounts program) and delete FILERB, FILERX and FILERT.

THE PAYMENTS FILE

Last month's article described the format of the file that you will need for holding details of your direct debits and standing orders. This must have the same name as your main bank file but be allocated to directory P. This 'Payments' file should be created using the main Filer program, and as stated in part one, it must contain the following fields in exactly the order given below.

Fieldname	Fieldwidth
DES	30
AMOUNT	8
FREQ	2
DATE1	6
DATE2	6

You can now use the main Filer program to add records to this file. In each case, the field DES should contain a suitable description of the direct debit, AMOUNT the amount of money payable, DATE1 the date of the first payment, and DATE2 the date of the last payment. The field FREQ should contain '1' if the payment is annual, and '2' if the payment is monthly. No other forms of payment are catered for. Dates should be entered in the form given last month (YYMMDD). For the last date, you can just give any date in the future (e.g. 999999 if you wish). Remember, too, to enter amounts as money sums, with a decimal point and two figures following.

CREATING BANK STATEMENTS

That's all there is to it. You will now find that when you run the Filer Accounts program and ask for a statement covering any particular period, all the direct debit and standing order payments that occur in that period will be automatically included in your statement (and taken into account when calculating a balance). It really is that simple.

Of course, you can always go back to Filer to edit your payments file, and to delete or add new direct debit payments. You could also use this file to handle any regular credits (maybe your salary for example) but you must specify this as a negative amount. I have been using this program personally for about two months now, and it certainly removes the tedium of my previous manual accounting. What's more, I can now be certain that the arithmetic is correct and that all regular payments are entered, something that was never true in the past!

PROGRAM NOTES

The technique used to handle direct debit and similar payments is as follows. When a statement or balance is called for,

the program reads each payment record in turn and for each calculates all the payment dates that will occur within the selected accounting period. These dates, together with pointers to the original payment records, are formed into an ordered linked list. This is held in an array which currently allows up to 20 payments in any one accounting period. You can increase this, if you wish, by changing the value assigned to np at line 1090.

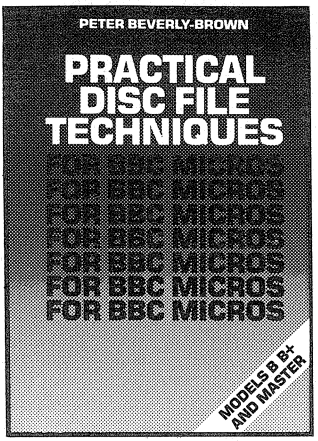
Once this list is complete, the statement is compiled, inserting any direct debits as the date arises by reading the description and amount from the payments file. All such payments are marked on the statement by an asterisk in the cheque field and always appear before any other transactions on the same date. You can change the asterisk if you wish in line 7840. You will also find that the first date on a statement cannot be earlier than that of the first record in your bank file, and that the last is the date of the last record in that file.

The program is, like the other Filer programs, written to assist those who may wish to modify it for their own needs, though several of the routines used in Filer Accounts are rather more complex than in earlier Filer programs. Copies of the notes on the main Filer database program are still available on receipt of an A5 SAE, and the complete Filer Accounts program is included on this month's magazine/cassette disc.

```
  160 PROCclose:VDU26,12:*FX4,0
 1060 debit%=FALSE:FDR=256
 1090 np=20:DIM pay(np,2),size(1),f(1)
 3180 size(0)=recs:f(0)=f:PROCpayments:E
NDPROC
 3460 debit%=-1:INPUT#F1,recp,I,size(1),
f(1)
 4000 DEF PROCbalance:LOCAL d2
 4030 PROCread(rec-1,F,0):d2=FNval(date)
 4040 IF rec>recb+1 PROCpay(db,d2):PROCs
heet(recb,db,d2,FALSE,FALSE)
 4320 PROCread(1,F,0):IF d1<FNval(date)
d1=FNval(date)
 4330 PROCread(rec-1,F,0):IF d2=0 OR d2>
FNval(date) d2=FNval(date)
 4350 PROCpay(d1,d2)
 4500 DEF PROCsheet(r,d1,d2,p1%,p2%)
 4590 IF pay>-1 THEN IF pay(pay,0)<=FNva
l(date) THEN PROCpay2:GOTO4590
 7000 DEF PROCpay(d1,d2):LOCAL I
 7020 pay=-1:fp=0:lp=-1
 7040 IF recp=1 ENDPROC
 7060 FOR I=1 TO recp-1:PROCread(I,F1,1)
:PROCcheck(d1,d2,VAL(record$(3,1)),I):NE
XT I
 7080 ENDPROC
 7100 :
 7200 DEF PROCcheck(f,r):LOCAL d,di
 7220 di=INT(10^(6-2*f)+0.5)
 7240 FOR d=VAL(record$(4,1)) TO VAL(rec
ord$(5,1)) STEP di
 7260 IF VAL(MID$(STR$(d),3,2))=13 d=d+8
800
 7280 IF d>=d1 AND d<=d2 PROClistp(d,r,p
ay)
 7300 NEXT d
 7320 ENDPROC
 7340 :
 7400 DEF PROClistp(d,r,s):LOCAL K%,s1
 7420 IF fp>np PRINT"Too many paymEnts":
ENDPROC
 7440 IF s=-1 THEN PROCadd(d,r,s,-1):END
PROC
 7460 IF d<=pay(s,0) THEN PROCadd(d,r,s,
-1):ENDPROC
 7480 K%=0:REPEAT:s1=s:s=pay(s,2)
 7500 IF s>-1 THEN IF d<pay(s,0) THEN K%
=-1
 7520 UNTIL s=-1 OR K%
 7540 PROCadd(d,r,s,s1)
 7560 ENDPROC
 7580 :
 7600 DEF PROCadd(d,r,s,s1)
 7620 IF s1=-1 THEN pay(fp,2)=s:pay=fp E
LSE pay(fp,2)=pay(s1,2):pay(s1,2)=fp
 7640 pay(fp,0)=d:pay(fp,1)=r:fp=fp+1
 7660 ENDPROC
 7680 :
 7800 DEF PROCpay2
 7820 PROCread(pay(pay,1),F1,1):balance=
balance-100*VAL(record$(2,1))
 7840 IF p2% PROCprintline(STR$(pay(pay,
0)),record$(1,1)," * ",FNstrip(record$(2
,1),"."),STR$(balance DIV100)+"."+RIGHT$
(STR$(balance),2))
 7860 pay=pay(pay,2)
 7880 ENDPROC
 7900 :
20700 DEF PROCread(n,C,R):LOCAL I
20720 PTR#C=FDR+size(R)*(n-1)
20740 FOR I=1 TO f(R):INPUT#C,record$(I,
R):NEXT I
20760 ENDPROC
20780 :
20800 DEF PROCwrite(n,C,R):LOCAL I
20820 PTR#C=FDR+size(R)*(n-1)
20840 FOR I=1 TO f(R):PRINT#C,record$(I,
R):NEXT I
20860 ENDPROC
```

Practical Disc-File Techniques

Good books on file handling are few and far between. Mike Williams, fresh from his endeavours with the BEEBUG Filer, reports on one which is enjoyable to read as well.



If you have been following our various articles on the BEEBUG Filer database program that have appeared in BEEBUG since last November (Vol.4 No.6) then you will have become familiar with the use of files and databases. You may have considered writing some file-handling programs yourself, but unless you are already a fairly competent programmer, the thought of doing so may have seemed quite daunting. Where do you start? And where can you find out all you may need to know?

Well, this new book by Peter Beverley-Brown may be just what you are looking for. The approach is very practical and down-to-earth, and ideal for the programmer who wants to extend his knowledge to file handling. The book has the major merit of being eminently readable, so much so that like a good novel you will find it hard to put it down. And yet, even the more experienced programmer is likely to find much of interest.

Initially, the book covers much of the essential detail required for file handling, how to create files of the right length, how to read and write to files, and ideas like that used in Filer, where a description of the files structure can itself be stored as part of the file.

Chapter 4 describes the programming of 'serial' files, though I have some reservations with the authors terminology. The point is that a disc is intrinsically a direct access device - so you can always proceeds directly to any location on the disc. But this is only a minor quibble, and the book is too full of good advice to worry unduly about academic niceties.

'Random' access is covered in the next chapter, but I would claim that the author is really describing a sequential file to which the disc drive provides direct access to each record. Again, ignoring the terminology, the information and examples given are useful and practical. Chapter 6 deals with the rudiments of index-sequential files which is an important technique for the fast access of records held in some kind of order. It is a file format commonly used commercially and well worth the trouble of learning about.

The chapter on searching and sorting is rather disappointing, despite several references by the author to past issues of BEEBUG. If the entire contents of a file can be read into the computer's memory then searching and sorting is easy. Dealing similarly with files that are much larger requires different techniques and these are only touched on briefly. This is an important subject because a poor choice of algorithm can easily lead to inefficient and slow file processing. More information here would have been welcome.

The book concludes with a round-up of disc-file hints for 'good house-keeping' and a complete listing of the author's 'Really Useful Program' for file handling. Despite my criticisms of some of the later sections of this book, I thoroughly recommend it to those BEEBUG members who wish to learn about file handling from scratch. In this respect it must be one of the most useful and readable books on this subject yet produced.

SPECIAL OFFER

We have been able to negotiate direct with Peter Beverley-Brown to offer his book to BEEBUG members at the price of just £4.95. Full details are contained in this month's supplement.

POSTBAG

POSTBAG

BEEBUG for Amstrad

I would like as a member of BEEBUG to have a discount on your other products. I have an Amstrad as well as a BBC micro and I may want the same programs for that as I use (and like) on my Beeb.

Brian Westbury

We are very happy to offer BEEBUG members the same 25% discount on our Amstrad products (for the CPC464, 664 & 6128) as on our software for the BBC micro. Just quote your membership number as usual to claim the discount.

If you want more information on our Amstrad range of software, just send us an A5 SAE for our price list.

Staggering News

There are two ways of formatting discs for the BBC micro. One method might be called 'straight' in which sector 0 of each track lies on the same radius. The other, which is 'staggered', has an offset between adjacent tracks of one sector. The latter increases the speed of reading and writing as less time is spent waiting when moving to the next track.

I have at my disposal 5 formatting 'programs'. Of these, 3 produce 'straight' formatting and 2 give a 'staggered format'. The Watford DFS with 62 file catalogue gives 'straight' formatting, so I have to balance the advantage of a 62 file catalogue against a decrease in speed.

However, if I use Disc Doctor to format, and then *BACKUP from the Watford formatted disc, (alternatively use *DZAP to rewrite the appropriate sectors on the Disc Doctored disc) I can enjoy the increased catalogue size of the Watford DFS and with the speed of Disc Doctor.

D.A.F.Driver

Very interesting, as they say. How much you might benefit from this obviously depends on how much disc accessing you do. It is another example of how efficient techniques in disc filing can save a lot of time.

Not Where it Seems

In the Master Series article "Using Sideways RAM" BEEBUG Vol.5 No.3, Thomas Nunns explains that the extra commands to access SWR are located in the Master's MOS. This is not so; like the B+ they are located in the DFS "for economy of space" to quote Reference Manual Part 1. Thus these commands will not be available if the DFS is not present. On the Master this will be so if ROM 9 (DFS position) is logically 'unplugged'.

I hope this clarifies the only grey area in a very good article.

A.M.Collyer

Clearly, knowledge of the location of the SWR code could be crucial, and our author, in advance of the Acorn reference manuals, made what seemed like a logical assumption.

Privates on Parade

The Master Series utility to save and load function key definitions in BEEBUG Vol.5 No.2 seemed somewhat cumbersome. The following short program will save a spooled file containing the function key definitions:

```
10 INPUT"Filename for defin
itions",f$
20 OSCLI"SPOOL "+f$
30 FOR C%=0 TO 9
40 PRINT"*KEY";C%;:OSCLI"SH
OW "+STR$C%
50 NEXT
60 *SPOOL
```

The resulting file may be *EXECed to load the function key definitions without corrupting anything except the keyboard buffer.

There are some minor limitations to this method. The format of SHOW limits the length of each key definition to 231 characters (in View the limit is 61 characters).

David Watkins

Thomas Nunns replies: Mr Watkins points are quite valid, and subject to some limitations, spooling a file of key definitions will often suffice. My article was as much an exercise in saving and loading techniques for the private RAM and could easily be adapted for any of the private RAM pages. Saving and loading an entire page directly also ensures that any 'illegal' use that may have been made of the area is precisely duplicated.

HINTS and tips HINTS and tips HINTS and tips HINTS and tips HINT and

More Markers in Wordwise

There is often a need for more than the two markers provided by Wordwise Plus. These three function key definitions will simulate another pair of markers. Pressing Shift/Ctrl f0 will replace the existing markers with '~' and return the cursor to its original position.

You can then use the markers 'again' as you need. If necessary, you can now press Shift/Ctrl f1 to remove these markers. Shift/Ctrl f2 will replace the '~' symbol with markers to restore the original state. The '~' symbol below can be replaced with another if you wish, but do include the spaces.

```
*KEY0 "^|!|O |!|O|!$|!#
|A~|!|O|!$|!#|A~|!|O|A|!$^|
A"
*KEY1 "^|!|O |!|O|!$|!#
|A|!|O|!$|!#|A|!|O|A|!$^|A"
*KEY2 "^|!|O |!|O|!$~|A
|!#|!|O|!$~|A|!#|!|O|A|!$^|
A"
```

D.P. Rendall

ROM Reminder

In these days of ROMs on disc for loading into sideways RAM and sideways ZIF sockets for readily removable ROMs, it is difficult to know whether a particular ROM is going to be there when a command is issued. With the 177ØDFS you can use *BUILD to create an ASCII file on disc with the same name as the ROM command. Include a REM statement (to avoid causing an error) to the effect that the ROM concerned is not present. This will be printed if the ROM is not able to accept the command. For example:

```
*BUILD WORD
ØØ1 REM View ROM not
plugged in
```

Colour Mixing

Although using values greater than 4 for the first GCOL parameter is one method of producing mixed colours on the screen, this is somewhat unpredictable. A better method for mode 2 is demonstrated in line 3Ø of this short program to produce an orange triangle of red and yellow stripes:

```
1Ø MODE2
2Ø firstcol=1:seccol=3
3Ø ?&359=((firstcol?&C42
A)AND&AA)+((seccol?&C42A)AN
D&55)
4Ø MOVE 1ØØ,1ØØ:MOVE 118
Ø,1ØØ:PLOT85,64Ø,9ØØ
```

This works on the model B, B+, and Master (but not on the Comapct or across the Tube).

Wordwise Plus Word Count

The word count displayed by Wordwise Plus is often inflated by any embedded commands present in the text. The following segment program, called from the edit mode (with, say, Shift-f5), returns a true word count from the cursor position down to the first occurrence of f1-@, if included, or the end of the text if not. After counting, the f1-@ is automatically deleted.

```
REM WORD COUNT
SELECT TEXT
FKEY5,"@"
V%=Ø
P."Recounting"
REPEAT
FIND "|G"
V%=V%+1
CURSOR RIGHT
IF GCT$="@" THEN CURSOR
BOTTOM
UNTIL EOT
DISPLAY
VDU 12,31,Ø,3
P."Net Count (Cursor to
@) = ";
P.W%-V%+1
VDU 31,Ø,23
P."Press Copy ";
CURSOR TOP
REPLACE "|G@",""
RECOUNT
CURSOR TOP
```

Jac Herberg

Listing Variables

The following short function key definition will list the names of all the real variables used by a program. Note that only those variables actually used (and assigned to) when the program was run will be listed.

```
*KEYØ F.F%=&482TO&4F4S.
2:A%=!F%A.&FFFF:IFA%>&FF RE
P.V.1Ø,13,-512+F%/2:M%=A%+1
:REP.M%=M%+1:V.?M%:UN.?M%=Ø
:A%=!A%A.&FFFF:UN.A%<&FF:N.
EL.N.|M
```

Bug in Scrolling

For an interesting (but useless) MOS bug in the Beeb's screen scrolling run the following short program.

```
1Ø MODE3:PRINTSTRING$(4
5,CHR$1Ø)
2Ø PRINTTAB(45,3)"Ø1234
56789"
3Ø VDU28,Ø,24,79,Ø
4Ø VDU3Ø,11,11,11
```

Ian Tresman

TUTANKHAMEN

Egyptology may not be high on your list of priorities but the lastest game from our acclaimed games writer, Jonathan Temple, will have you cavorting in the catacombs. Without doubt, another outstanding game.

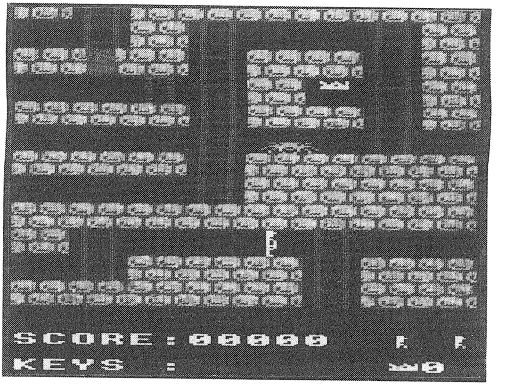
As an intrepid explorer, you have become lost in the maze-like tombs of Tutankhamen. Deciphering a strange message inscribed in the stonework, you discover that before the Gods will let you leave you must fulfil an ancient prophecy. This entails collecting the sixteen gold crowns scattered throughout the pyramid.

The Egyptians have included many traps in the sixteen chambers of the pyramid, to foil even the most skilful adventurers. There are boulders, which are dislodged as you pass under them and fall, blocking your way back; there are locked doors, and corresponding coloured keys to find; and last, but by no means least, are the deadly spiders that inhabit the chambers.

To escape from the pyramid you will need to collect all sixteen crowns and find the key to the last door. You will probably need to make a map as you go so that you can find your way around more easily. Once you have completed the game you will be awarded a large bonus, depending on how long you take and how many lives you have remaining.

The keys to use are 'Z' and 'X' for left and right movement and '*' and '?' to go up or down a ladder. In addition the game can be paused by pressing 'P', and continued with 'C'.

While the game is paused 'Q' and 'S' will turn the sound on and off respectively and 'R' will allow you to return to the starting point - useful if you become trapped, but you do lose a life.



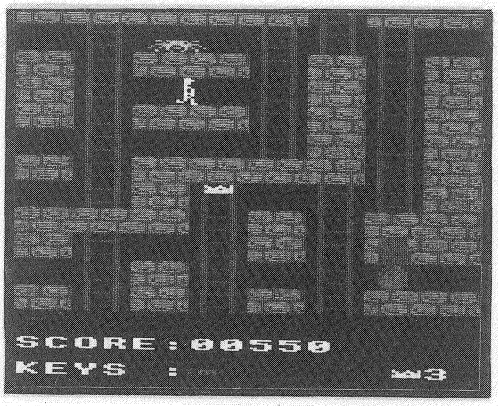
The program should be fairly straightforward to enter, although the data for the sixteen screens (lines 2830-3160) and the doors and keys (lines 2300-2390) should be entered carefully. Also, since the program runs in mode 2, those users with systems where PAGE is over &E00 will need to include a few lines at the start of the program to move it down in memory:

```
    1 IF PA.<&E01 THEN 10
    2 *K.0 *T.|MF.A%=0TO(TOP-PA.)S.4:A%!&E
00=A%!PA.:N.|MPA.=&E00|MO.|MDEL.1,4|MRUN|M
    3 *FX 138,0,128
    4 END
```

```
   10 REM PROGRAM TUTANKHAMEN
   20 REM VERSION B0.1
   30 REM AUTHOR  J. Temple
   40 REM BEEBUG  OCTOBER 1986
   50 REM PROGRAM SUBJECT TO COPYRIGHT
   60 :
  100 ON ERROR GOTO 3740
  110 MODE 2
  120 PROCtitle
  130 PROCinit
  140 PROCchars
  150 PROCenvs
  160 REPEAT
  170 PROCsetvars
  180 REPEAT
  190 PROCreset
  200 PROCscreen(P%)
  210 REPEAT
  220 PROCman
  230 PROCspider
  240 UNTIL E%
  250 IF E%=1 PROCkilled
```

```
  260 UNTIL Z%=0 OR E%=2
  270 IF E%=2 PROCcongrats ELSE PROCend
  280 UNTIL FALSE
  290 :
 1000 DEFPROCman
 1010 A%=X%:B%=Y%:C%=V%:D%=W%
 1020 IFINKEY-56 PROCpause
 1030 IFF% IFPOINT(X%+24,Y%+8)<>14 IFX%
MOD64=32 PROCfall
 1040 IFINKEY-98 IFPOINT(X%-8,Y%-36)MOD
15=0 IFY%MOD64=28 X%=X%-32:W%=W% EOR1:S
OUND18,-10,50,1:IFV%<>233 V%=233:W%=236
 1050 IFINKEY-67 IFPOINT(X%+64,Y%-36)MO
D15=0 IFY%MOD64=28 X%=X%+32:W%=W% EOR1:
SOUND18,-10,50,1:IFV%<>232 V%=232:W%=23
4
 1060 IFINKEY-73 IFPOINT(X%,Y%+8)=15 IF
X%MOD128=32 Y%=Y%+32:V%=V% EOR1:W%=W% E
OR1:SOUND18,-10,75,1:IFC%<>238 V%=238:W
%=240
 1070 IFINKEY-105 IFPOINT(X%,Y%-68)=15
IFX%MOD128=32 Y%=Y%-32:V%=V% EOR1:W%=W%
 EOR1:SOUND18,-10,75,1:IFC%<>238 V%=238
:W%=240
 1080 IFD%<>W% GCOL3,7:MOVE A%,B%:VDUC%
,10,8,D%:MOVE X%,Y%:VDUV%,10,8,W%
 1090 IFPOINT(X%+8,Y%-24) IFPOINT(X%+8,
Y%-24)<6 PROCkey
 1100 IFPOINT(X%+8,Y%-20)=12 IFX%MOD64=
32 PROCcrown
 1110 IFPOINT(X%+24,Y%+8)=14 F%=TRUE
 1120 IFX%<128 OR X%>1088 OR Y%<316 OR
Y%>956 PROCnew
 1130 ENDPROC
 1140 :
 1150 DEFPROCfall
 1160 R%(P%-1,2)=1:SOUND0,1,150,1
 1170 GCOL3,10
 1180 MOVE R%(P%-1,0)*128+32,1052-R%(P%
-1,1)*64
 1190 VDU229,10,8,230:PLOT0,-64,-32
 1200 VDU229,10,8,230:F%=0
 1210 ENDPROC
 1220 :
 1230 DEFPROCnew
 1240 IFF% R%(P%-1,2)=1:F%=0
 1250 IFX%>1088 P%=P%+1:X%=128
 1260 IFX%<128  P%=P%-1:X%=1088
 1270 IFY%>956  P%=P%-(G%*2-2):G%=G%-1:
Y%=316
 1280 IFY%<316  P%=P%+G%*2:G%=G%+1:Y%=9
56
 1290 IFP%=17 E%=2 ELSE PROCscreen(P%)
 1300 ENDPROC
 1310 :
 1320 DEFPROCkey
 1330 L%=-1:REPEAT L%=L%+1
 1340 UNTIL K%(L%,0)=P%
 1350 SOUND17,2,75,6:K%(L%,3)=1
 1360 MOVE X%,Y%:GCOL3,L%+1:VDU242
 1370 D%(L%,3)=0:PROCscore(250)
 1380 ENDPROC
 1390 :
 1400 DEFPROCcrown
 1410 C%(P%-1,2)=1:SOUND17,2,100,3
 1420 MOVE X%,Y%:GCOL3,12:VDU231
 1430 H%=H%+1:PROCscore(100)
 1440 ENDPROC
 1450 :
 1460 DEFPROCspider
 1470 IFABS((R%+16)-Y%)<48 IFABS((Q%+32
)-X%)<96 E%=1
 1480 GCOL3,2:MOVE Q%,R%:VDU245,246
 1490 Q%=Q%+SX%:R%=R%+SY%
 1500 MOVE Q%,R%:VDU245,246
 1510 IFSX% IFQ%=S1% OR Q%=S3% SX%=-SX%
 1520 IFSY% IFR%=S2% OR R%=S4% SY%=-SY%
 1530 IFABS((R%+16)-Y%)<48 IFABS((Q%+32
)-X%)<96 E%=1
 1540 ENDPROC
 1550 :
 1560 DEFPROCpause
 1570 I=TIME:REPEAT N%=GET AND &DF
 1580 IF N%=81 THEN *FX 210,1
 1590 IF N%=83 THEN *FX 210,0
 1600 UNTIL N%=67 OR N%=82
 1610 TIME=I:IF N%=82 E%=1
 1620 ENDPROC
 1630 :
 1640 DEFPROCscore(N%)
 1650 S%=S%+N%:VDU4,17,128,17,7,31,8,27
 1660 PRINT LEFT$("000000",5-LEN(STR$(S%
)))+STR$(S%)
 1670 PRINTTAB(16,29);H%;TAB(8,29);
 1680 FOR L%=0 TO 4
 1690 IFK%(L%,3) VDU17,L%+1,242
 1700 NEXT:VDU5
 1710 ENDPROC
 1720 :
 1730 DEFPROCkilled
 1740 Z%=Z%-1:SOUND 0,1,100,2
 1750 FOR N%=1 TO 2000:NEXT
 1760 ENDPROC
 1770 :
 1780 DEFPROCreset
 1790 P%=3:G%=2
 1800 X%=672:Y%=412:V%=232:W%=234
 1810 FOR L%=0 TO 15
 1820 R%(L%,2)=0:NEXT
 1830 ENDPROC
 1840 :
 1850 DEFPROCend
 1860 *FX 15,0
 1870 VDU4,28,4,16,15,9,17,128,12,26,5
 1880 PROCprint("GAME OVER",288,700,1,3
)
 1890 PROCprint("<SPACE>",352,572,4,6)
 1900 REPEAT UNTIL GET=32
 1910 ENDPROC
 1920 :
```

```
1930 DEFPROCprint(T$,X,Y,A,B)
1940 LOCAL A%,N%,X%,Y%
1950 X%=&70:Y%=0:A%=10
1960 FOR N%=1 TO LEN(T$)
1970 ?&70=ASC(MID$(T$,N%)):CALL&FFF1
1980 VDU23,254,?&71,?&71,?&72,?&72,?&7
3,?&73,?&74,?&74
1990 VDU23,255,?&75,?&75,?&76,?&76,?&7
7,?&77,?&78,?&78
2000 MOVE X+N%*64,Y:GCOL 0,A
2010 VDU254,10,8,255
2020 MOVE X+N%*64-8,Y-4:GCOL 0,B
2030 VDU254,10,8,255
2040 NEXT
2050 ENDPROC
2060 :
2070 DEFPROCsetvars
2080 Z%=3:S%=0:P%=3:G%=2:H%=0:TIME=0
2090 X%=672:Y%=412:V%=232:W%=234
2100 VDU19,15,4;0;19,8,7;0;19,13,2;0;
2110 VDU19,12,3;0;19,10,1;0;19,14,1;0;
2120 FOR L%=0 TO 4
2130 K%(L%,3)=0:D%(L%,3)=1:NEXT
2140 FOR L%=0 TO 15
2150 C%(L%,2)=0:NEXT
2160 ENDPROC
2170 :
2180 DEFPROCinit
2190 DIM K%(4,3),D%(4,3),R%(15,2),C%(1
5,2)
2200 FOR L%=0 TO 4:FOR M%=0 TO 2
2210 READ K%(L%,M%):NEXT,
2220 FOR L%=0 TO 4:FOR M%=0 TO 2
2230 READ D%(L%,M%):NEXT,
2240 FOR L%=0 TO 15
2250 READ R%(L%,0),R%(L%,1):NEXT
2260 FOR L%=0 TO 15
2270 READ C%(L%,0),C%(L%,1):NEXT
2280 ENDPROC
2290 :
2300 DATA 6,6,4,1,7,11,9,7,2
2310 DATA 16,6,8,10,2,11
2320 DATA 3,3,4,8,7,4,14,8,4
2330 DATA 11,3,11,16,6,11
2340 DATA 6,1,8,5,2,3,6,10,5,1,7,10
2350 DATA 7,10,5,3,2,1,7,3,3,3,3,10
2360 DATA 7,8,7,10,7,10,4,1
2370 DATA 4,11,3,2,6,4,7,2,5,4,3,2
2380 DATA 4,8,1,2,2,4,4,4,3,6,7,10
2390 DATA 2,10,4,11,2,6,6,4
2400 :
2410 DEFPROCscreen(P%)
2420 VDU4,17,128,12,23;10,32;0;0;0;17,
7
2430 PRINTTAB(2,27);"SCORE:";TAB(15,27
);STRING$(Z%-1,CHR$232+" ");TAB(2,29);"
KEYS :";
2440 PROCscore(0)
2450 VDU4,17,3,31,15,29,231,17,132
2460 RESTORE2830:IFP%>1 PROCread
2470 READ C%,S1%,S2%,S3%,S4%,SX%,SY%,W$
2480 FOR L%=0 TO 11
2490 A%=EVAL("&"+MID$(W$,L%*2+1,2))
2500 FOR N%=0 TO 7
2510 VDU31,N%*2+2,L%*2+1
2520 IF (A% AND(2^N%)) VDU17,C%,224,22
4,10,8,8,225,226,11
2530 NEXT,
2540 VDU5,18,3,15
2550 IFP%=17 THEN ENDPROC
2560 FOR L%=256 TO 896 STEP 128
2570 FOR N%=284 TO 988 STEP 64
2580 IFPOINT(L%,N%)=0 IFPOINT(L%,N%-68
)=0 OR POINT(L%,N%+4)=0 MOVE L%,N%:VDU2
27,228,10,8,8,227,228
```




```
2590 NEXT,
2600 FOR L%=0 TO 4:GCOL 0,L%+1
2610 IFK%(L%,0)=P% IFK%(L%,3)=0 MOVE K
%(L%,1)*128+32,1052-K%(L%,2)*64:VDU 242
2620 IFD%(L%,0)=P% IFD%(L%,3)=1 OR (L%
=4 AND H%<16) MOVE D%(L%,1)*128+32,1052
-D%(L%,2)*64:VDU 243,10,8,244
2630 NEXT
2640 MOVE R%(P%-1,0)*128+32,1052-R%(P%
-1,1)*64
2650 GCOL 0,4:PLOT0,56,0:PLOT81,0,-60
2660 PLOT0,-56,0:PLOT81,0,60
2670 IFR%(P%-1,2)=1 PLOT0,0,-64
2680 GCOL 3,10:VDU229,10,8,230
2690 IFC%(P%-1,2)=0 MOVE C%(P%-1,0)*12
8+32,1052-C%(P%-1,1)*64:GCOL 3,12:VDU23
1
2700 GCOL 3,7:MOVE X%,Y%:VDUV%,10,8,W%
2710 S1%=S1%*128:S2%=1020-S2%*64
2720 S3%=S3%*128:S4%=1020-S4%*64
2730 E%=0:Q%=S1%:R%=S2%:GCOL3,2
2740 MOVEQ%,R%:VDU245,246
2750 ENDPROC
2760 :
2770 DEFPROCread
2780 FOR L%=1 TO P%-1
```

```
 2790 READ C%,S1%,S2%,S3%,S4%,SX%,SY%,W
$
 2800 NEXT
 2810 ENDPROC
 2820 :
 2830 DATA 1,5,3,5,11,0,-32
 2840 DATA FF81ADADADADADAD8DADA5FD
 2850 DATA 2,2,2,2,6,0,-16
 2860 DATA FF09ED21BD01FF01F5C511DD
 2870 DATA 2,2,6,7,6,32,0
 2880 DATA FD84B790B700F7F0FF01DCDF
 2890 DATA 2,4,2,4,11,0,-32
 2900 DATA FFB1B5B5B5B485B585F485BD
 2910 DATA 5,2,2,2,9,0,-32
 2920 DATA FF81BD21FDF105BD81FB0BAB
 2930 DATA 5,4,6,7,6,24,0
 2940 DATA DD01F7D0DF05F417D1DD04F5
 2950 DATA 5,2,2,7,2,32,0
 2960 DATA DF00ADA1ADA0BD84D7D104D5
 2970 DATA 5,2,6,7,6,32,0
 2980 DATA BD04F701F581B5B1B781B4B7
 2990 DATA 5,2,5,7,5,16,0
 3000 DATA FFB087BC81DFD195F5C1D5D5
 3010 DATA 6,6,2,6,6,0,-16
 3020 DATA FFC1DD11DF81BD85B58531FF
 3030 DATA 6,7,2,7,10,0,-32
 3040 DATA ABA3BF80B721ADA1B717C0FF
 3050 DATA 6,2,6,6,6,32,0
 3060 DATA F515D515F5C0D501F794C1FF
 3070 DATA 6,4,2,4,11,0,-32
 3080 DATA D514D7D4C515F5C017F107FF
 3090 DATA 6,2,6,7,6,32,0
 3100 DATA B784D515D500F5C5DCDD10FF
 3110 DATA 6,6,2,6,9,0,-32
 3120 DATA D505D5C417D405DD05F504FF
 3130 DATA 6,2,2,2,11,0,-32
 3140 DATA FF80AD85F485AC85F4F500FF
 3150 DATA 5,0,0,0,0,0,0
 3160 DATA FF0000FF00000000FFFFFFFF
 3170 :
 3180 DEFPROCcongrats
 3190 PROCscreen(17):PROCscore(Z%*8000)
 3200 IFTIME<60000 PROCscore((60000-TIM
E)*3)
 3210 PROCprint("Congratulations!",64,8
96,1,3)
 3220 PROCprint("You have escaped",64,7
04,4,6)
 3230 PROCprint("from the Pyramid",64,6
08,4,6)
 3240 RESTORE 3320:N%=81:*FX 15,0
 3250 FOR L%=1 TO 10:READ A%,D%:N%=N%+A%
 3260 SOUND 1,-10,N%,D%
 3270 SOUND 2,-5,N%+48,D%:NEXT
 3280 REPEAT UNTIL ADVAL(-7)=15
 3290 FOR I=1 TO 10000:NEXT
 3300 ENDPROC
 3310 :
 3320 DATA 0,4,8,4,8,4,4,4,8,8,-12,8,4,
8,-12,8,8,8,-16,8
 3330 :
 3340 DEFPROCtitle
 3350 VDU5
 3360 PROCprint("The Tombs of",224,832,
1,3)
 3370 PROCprint("TUTANKHAMEN",256,736,4
,6)
 3380 PROCprint("Await you",328,640,1,3
)
 3390 PROCprint("<SPACE>",384,352,4,5)
 3400 REPEAT UNTIL GET=32
 3410 ENDPROC
 3420 :
 3430 DEFPROCchars
 3440 VDU23,224,124,-2,-2,190,190,198,1
24,0
 3450 VDU23,225,227,247,247,245,245,118
,227,0
 3460 VDU23,226,227,247,247,245,245,54,
227,0
 3470 VDU23,227,24,24,24,24,31,24,24,24
 3480 VDU23,228,24,24,24,24,248,24,24,2
4
 3490 VDU23,229,24,44,94,94,191,191,191
,-1
 3500 VDU23,230,-1,-1,-1,-1,-1,127,126,
60
 3510 VDU23,231,0,137,221,119,-1,-1,0,0
 3520 VDU23,232,48,56,48,48,32,48,40,40
 3530 VDU23,233,12,28,12,12,4,12,20,20
 3540 VDU23,234,40,56,40,48,32,32,32,48
 3550 VDU23,235,40,56,40,48,40,168,200,
12
 3560 VDU23,236,20,28,20,12,4,4,4,12
 3570 VDU23,237,20,28,20,12,20,21,19,48
 3580 VDU23,238,24,26,26,126,88,88,24,2
8
 3590 VDU23,239,24,88,88,126,26,26,24,5
6
 3600 VDU23,240,20,20,20,22,16,16,16,48
 3610 VDU23,241,40,40,40,104,8,8,8,12
 3620 VDU23,242,0,0,96,159,149,149,96,0
 3630 VDU23,243,-1,195,0,195,0,195,0,19
5
 3640 VDU23,244,0,195,0,195,0,195,0,195
 3650 VDU23,245,0,55,11,63,78,149,36,34
 3660 VDU23,246,0,236,208,-4,114,169,36
,68
 3670 ENDPROC
 3680 :
 3690 DEFPROCenvs
 3700 ENVELOPE 1,1,0,0,0,0,0,0,90,-1,-2
,-3,97,97
 3710 ENVELOPE 2,133,8,4,8,3,1,1,126,0,
0,-10,126,0
 3720 ENDPROC
 3730 :
 3740 MODE7:REPORT:PRINT" at line ";ERL
 3750 *FX15,1
 3760 END
```

BEEBUG MAGAZINE is produced by BEEBUG Publications Ltd.
Editor: Mike Williams
Technical Assistant: Alan Webster
Production Assistant: Yolanda Turuelo
Secretary: Debbie Sinfield
Managing Editor: Lee Calcraft
Additional thanks are due to Sheridan Williams, Adrian Calcraft, Geoff Bains, John Yale and Tim Powys-Lybbe.

Editorial Address
BEEBUG
Dolphin Place
Holywell Hill
St. Albans
Herts. AL1 1EX

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet, 'Notes of Guidance for Contributors', is available on receipt of an A5 (or larger) SAE.

In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using "Wordwise", "View", or other means, but please ensure an adequate written description of your contribution is also included. If you use cassette, please include a backup copy at 300 baud.

HINTS

There are prizes of £5 and £10 for the best hints each month, plus one of £15 for a hint or tip deemed to be exceptionally good.

Please send all editorial material to the editorial address above. If you require a reply it is essential to quote your membership number and enclose an SAE.

MEMBERSHIP RATES

Send all applications for membership, membership renewals, membership queries and orders for back issues to the membership address. All membership fees, including overseas, should be pounds sterling drawn (for cheques) on a UK bank.

MEMBERSHIP SUBSCRIPTION RATES

£ 6.90 6 months (5 issues) UK ONLY
£12.90 - 1 year (10 issues) UK, Eire, BFPO
£19.00 - Rest of Europe £22.00 Middle East
£24.00 Americas & Africa, £26.00 Elsewhere

BACK ISSUES
(Members only)

Vol	Single issues	Volume sets (10 issues)
1	90p	£8
2	£1	£9
3	£1.20	£10
4	£1.20	£11
5	£1.30	–

Please add the cost of post and packing as shown:

DESTINATION	First issue	Each subsequent issue
UK	30p	10p
Europe	70p	20p
Elsewhere	£1.50	50p

All overseas items are sent airmail (please send a sterling cheque). We will accept official UK orders but please note that there will be a £1 handling charge for orders under £10 that requires an invoice. Note that there is no VAT on magazines.

Back issues are for members only, so it is ESSENTIAL to quote your membership number with your order. Please note that the BEEBUG Reference Card and BEEBUG supplements are not supplied with back issues.

Membership, Back Issues & Software Address

BEEBUG
Dolphin Place
Holywell Hill
St. Albans
Herts. AL1 1EX

Hotline for queries and software orders

St. Albans (0727) 40303
Manned Mon-Fri 9am-4.30pm

24hr Answerphone Service for Access and Barclaycard orders, and subscriptions
St. Albans (0727) 40303

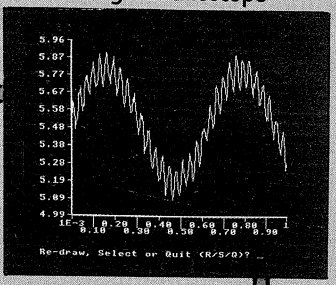
If you require members discount on software it is essential to quote your membership number and claim the discount when ordering.

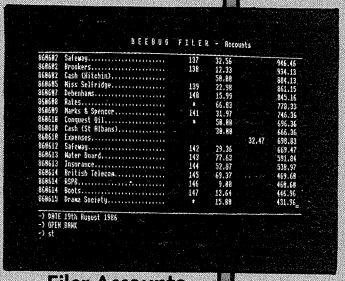
CASSETTE/DISC CONTENTS OCTOBER 1986

BASIC PROGRAM COMPARATOR — sort out your programs with this helpful utility.
STORAGE OSCILLOSCOPE — capture and display data with this useful program, complete with sample data sets.
THE MASTER SERIES — our superb ADFS menu program extended for Wordwise and View.
FILER ACCOUNTS — the complete accounts program to handle direct debits and standing orders just as easily as other everyday transactions.
PRINTER GRAPHICS — an easily adapted flexible screen dump for your printer.
BEEBUG WORKSHOP — complete demonstration of the date handler routines.
WHICH MACHINE? — let your programs work out automatically which system they are running on; and it detects the new Master Compact.
FIRST COURSE — useful games routine for detecting keyboard input with INKEY.
DYNAMICALLY CALLED FUNCTIONS — demonstration of this useful programming technique.
TUTANKHAMEN — nothing less than another superb Jonathan Temple game.
BEEBUGSOFT FORUM — all the routines listed this month.

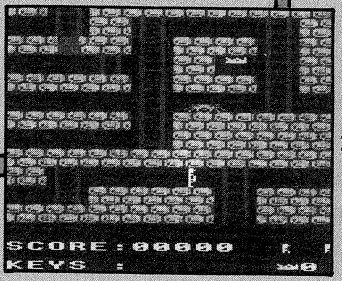
EXTRA FEATURES THIS MONTH
MAGSCAN — data for this issue of BEEBUG (Vol. 5 No. 5).

Storage Oscilloscope





Filer Accounts



Tutankhamen

All this for £3.00 (cass) £4.75 (disc) +50p p&p.
Back issues (disc since Vol. 3 No. 1, cass since Vol. 1 No. 10) available at the same prices.

Subscription rates	DISC UK	CASS UK	DISC O'seas	CASS O'seas
6 months (5 issues)	£25.50	£17	£30	£20
12 months (10 issues)	£50	£33	£56	£39

Prices are inclusive of VAT and postage as applicable. Sterling only please.

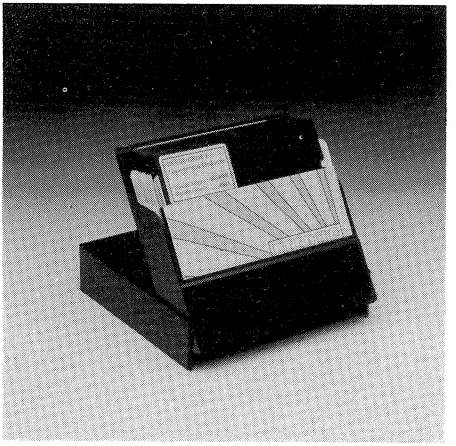
Cassette subscriptions can be commuted to disc subscription on receipt of £1.70 per issue of the subscription left to run.

All subscription and individual orders to
BEEBUG, Dolphin Place, Holywell Hill, St. Albans, Herts. AL1 1EX

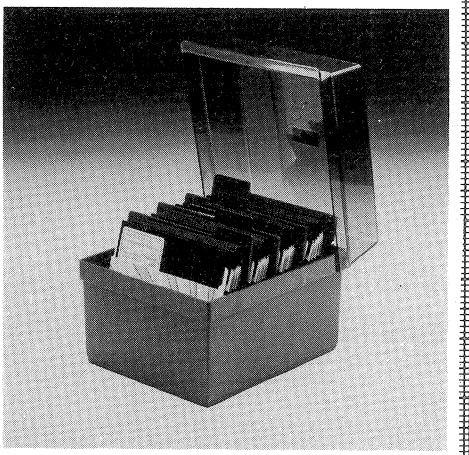
DYNAMIC DISCS

TESTED BY BEEBUG

NEW LOW PRICES



- ★ More than £30 off 50 96 tpi d/s
- ★ We are pleased to pass on a reduction in price from the manufacturer
- ★ Same High Quality
- ★ Leading UK Manufacturer
- ★ Guaranteed for Life by Beebug
- ★ Each Disc individually tested



- ★ Prices include UK delivery & VAT
- ★ Supplied in Free Storage Box
- ★ Official Orders Welcome
- ★ A further 5% discount for Beebug Members

48 TPI DOUBLE DENSITY

			MEMBERS				MEMBERS
10	S/S	£10.95	£10.40	10	D/S	£11.47	£10.90
25	S/S	£26.95	£25.60	25	D/S	£30.42	£28.90
50	S/S	£51.95	£49.35	50	D/S	£53.95	£51.25

96 TPI QUAD DENSITY

			MEMBERS				MEMBERS
10	S/S	£11.47	£10.90	10	D/S	£11.95	£11.35
25	S/S	£30.42	£28.90	25	D/S	£32.95	£31.30
50	S/S	£53.95	£51.25	50	D/S	£56.95	£54.10

ALL PRICES INCLUDE VAT & UK CARRIAGE

ACCESS/VISA orders welcome 0727 40303 (24 hours)
Beebug, Dolphin Place, Holywell Hill, St. Albans, Herts, AL1 1EX